Hans Heinrich Gloistehn

Programmieren von Taschenrechnern 3

Lehr- und Übungsbuch für den TI-58 und TI-59

TEXAS INSTRUMENTS

Mathematische Zeichen und Abkürzungen

$\mathbb{N}$	Menge der natürlichen Zahlen
$\mathbb{N}_0$	Menge der natürlichen Zahlen einschl. 0
$\mathbb{N}_k$	Menge der ersten k natürlichen Zahlen
$\mathbb{N}_{0,k}$	Menge der ersten k natürlichen Zahlen einschl. 0
$\in$	ist Element von
$\notin$	ist nicht Element von
$\wedge$	und
$\vee$	oder
*	steht für Prafixtaste [2nd], z.B. [*EXC] statt [2nd] [EXC / RCL]
AR	Anzeigeregister, im Text auch: Sichtfenster des Rechners
(AR)	Inhalt des AR; die im Sichtfenster angezeigte Zahl
R_n	Datenspeicher (Datenregister) mit der Adresse $n \in \mathbb{N}_{0,9}$
(R_n)	Inhalt des Datenspeichers R_n
T	Austauschspeicher, Vergleichsspeicher
(T)	Inhalt des T-Speichers
$(R_n) \rightarrow$ AR	Inhalt des R_n wird ins AR gebracht
(AR) $\rightarrow R_n$	Inhalt des AR wird in R_n gebracht
(AR) $\leftrightarrow (R_n)$	Austausch der Inhalte des AR und R_n
:=	Ergibt-Zeichen; gelesen: ‚wird ersetzt durch' oder ‚ergibt sich aus'
BAR	Befehlsadreßregister
BARS	Befehlsadreßregisterstelle
PSS	Programmspeicherstelle

Hans Heinrich Gloistehn

Programmieren von Taschenrechnern 3

**Lehr- und Übungsbuch
für den TI-58 und TI-59**

Vieweg

Vorwort

In den letzten Jahren haben die wissenschaftlichen Taschenrechner die bisherigen Rechenhilfsmittel (Rechenschieber, elektromechanische Tischrechner usw.) weitgehend abgelöst. Seit einiger Zeit befinden sich *programmierbare* Taschenrechner auf dem Markt, die wegen der vielfältigen Einsatzmöglichkeiten und des günstigen Preises ebenfalls rasch einen immer größer werdenden Abnehmerkreis finden werden. Diese Geräte können noch mehr als die bisherigen den Benutzer von lästiger und langwieriger Rechenarbeit befreien.

Dieses Buch wendet sich hauptsächlich an den im Programmieren unerfahrenen Leser, der seine ersten Kenntnisse auf diesem Gebiet mit einem programmierbaren Taschenrechner erwerben möchte. Wir denken dabei an die Studenten der ersten Semester naturwissenschaftlicher oder technischer Fachrichtung an Fachhochschulen oder Universitäten. Weiter an die Lehrer und Schüler der Sekundarstufe II. Der programmierbare Taschenrechner entlastet ja nicht nur vom aufwendigen Rechnen, sondern er erzieht auch zum logischen Denken und genauen mathematischen Formulieren. Schließlich denken wir auch an diejenigen, die ohne Nutzanwendung sich einfach nur aus Spaß und Freude mit dem Programmieren eines Taschenrechners beschäftigen möchten.

Der I. Teil des Buches ist nach Gesichtspunkten der Programmiertechnik geordnet. Es wird an Beispielen erklärt, wie der Taschenrechner zu programmieren ist. Die mathematischen Voraussetzungen sind dabei nicht sehr groß, Schulkenntnisse der 11./12. Klasse dürften ausreichend sein. Die Übungen am Ende jedes Abschnitts sollen dem Leser Gelegenheit geben, sein Wissen durch selbständiges Lösen von Aufgaben zu überprüfen. Die Lösungen sind im Anhang zusammengestellt. Im II. Teil des Buches findet der Leser Beispiele aus der Mathematik und Technik, die sich vorteilhaft mit einem programmierbaren Taschenrechner lösen lassen oder die programmiertechnisch interessant sind. Eine Vollständigkeit in der Erfassung der behandelten Gebiete wurde hier nicht angestrebt.

Im vorliegenden Band 3 der Reihe *Programmieren von Taschenrechnern* werden die Geräte TI-58 und TI-59 von Texas Instruments beschrieben. Auf die Behandlung der sehr interessanten und für den Praktiker außerordentlich nützlichen Modultechnik habe ich in diesem Buch verzichtet. Der Umfang des Buches wäre sonst noch stärker angewachsen. Der Benutzer eines programmierbaren Taschenrechners TI-58 oder TI-59 wird aus dem Handbuch von Texas Instruments leicht ersehen, wie Modul-Programme zu benutzen sind.

Den Mitarbeitern im Verlag Vieweg sei herzlich gedankt für die gute Zusammenarbeit beim Entstehen dieses Buches.

H. H. Gloistehn

Hamburg, im Winter 1977/78

Inhaltsverzeichnis

II. Teil: Anleitung zum Programmieren mit dem TI-58 und TI-59 1

1. Manuelles Rechnen 1
1.1. Die 4 Grundrechenarten 1
1.2. Termberechnungen 3
1.3. Datenspeicher (Datenregister, Memories) 5
1.4. Funktionstasten 10
1.5. Übungsaufgaben 13

2. Programmaufbau und Programmherstellung 14
2.1. Ein einfaches Programm entsteht 14
2.2. Das Starten und Anhalten eines Programms 16
2.3. Überprüfen und Korrigieren eines Programms 19
2.4. Programmadreßtasten 22
2.5. Indirekte Speicheradressierung 28
2.6. Die unbedingte Sprunganweisung GTO 31
2.7. Übungsaufgaben 36

3. Verzweigungen (bedingte Sprunganweisung) 37
3.1. Das Flußdiagramm für Verzweigungen 37
3.2. Der T-Speicher (das T-Register) 40
3.3. Mehrfache Verzweigungen 42
3.4. Verminderung und Überspringen bei Null 47
3.5. Flags (Flaggen, Signale) 53
3.6. Übungsaufgaben 57

4. Unterprogramme 59
4.1. Unterprogramme zur Berechnung von Funktionswerten 59
4.2. Unterprogramme als Zusammenfassung einer Folge gleicher Anweisungen 67
4.3. Unterprogramme höherer Stufe 68
4.4. Übungsaufgaben 74

5. Der Drucker PC-100 A 75
5.1. Auflisten eines Programms und der Datenspeicher 76
5.2. Das Ausdrucken von Daten 77

II. Teil: Programmbeispiele aus der Mathematik und Technik 81

6. Beispiele aus der Mathematik 81
6.1. Zahlen 81
6.1.1. Pythagoräische Zahlentripel 81

6.1.2. Größter gemeinsamer Teiler ... 82
6.1.3. Binomialkoeffizienten ... 84
6.2. Berechnung des ebenen Dreiecks ... 85
6.2.1. Cosinussatz ... 85
6.2.2. Sinussatz ... 88
6.3. Berechnung von Funktionswerten ... 90
6.3.1. Horner-Schema ... 90
6.3.2. Maximum einer Funktion ... 92
6.3.3. Besselsche Funktionen ... 95
6.4. Nullstellen von Funktionen (Gleichungen) ... 97
6.4.1. Iterationsverfahren ... 98
6.4.2. Newtonsches Verfahren ... 101
6.5. Numerische Integration ... 102
6.5.1. Trapezregel ... 103
6.5.2. Simpsonregel ... 105
6.5.3. Beispiele ... 106
6.6. Differentialgleichungen ... 107
6.6.1. Verbessertes Polygonzugverfahren ... 107
6.6.2. Differenzenschemaverfahren: Trapezregel ... 109
6.6.3. Beispiel ... 111

7. Beispiele aus der Technik ... 112
7.1. Stereostatik ... 112
7.1.1. Kräfte in der Ebene am Punkt ... 112
7.1.2. Auflagerkräfte beim Balken ... 114
7.2. Elastostatik (Festigkeitslehre) ... 117
7.2.1. Flächenmomente ... 117
7.2.2. Maximales Biegemoment ... 120
7.3. Kinematik / Kinetik ... 123
7.3.1. Kurbeltrieb ... 123
7.3.2. Bahnkurve ... 127
7.3.3. Bewegung einer Rakete ... 130
7.4. Elektrotechnik ... 133
7.4.1. Spannungsteiler ... 133
7.4.2. Berechnung eines Wechselstromkreises ... 136
7.4.3. Harmonische Analyse ... 138

8. Anhang ... 141
8.1. Tastenkode-Tabelle ... 141
8.2. Lösungen der Übungsaufgaben ... 141

Sachwortverzeichnis ... 149

I. Teil:
Anleitung zum Programmieren mit dem TI-58 und TI-59

1. Manuelles Rechnen

Wenn Sie das Programmieren mit einem Taschenrechner (im folgenden meistens kurz **Rechner** genannt) lernen wollen, so ist zunächst einmal eine sichere Beherrschung des *manuellen* Rechnens erforderlich. Viele von Ihnen benutzen vielleicht schon seit einiger Zeit einen technisch-wissenschaftlichen Rechner, anderen ist bisher nur die Handhabung eines einfachen Rechners mit Standardoperationen (+, −, ×, ÷) geläufig. Schließlich werden einige von Ihnen bisher überhaupt keine Erfahrung mit Rechnern besitzen. Wir stellen daher in diesem 1. Kapitel die Grundbegriffe des manuellen Rechnens zusammen. Überprüfen Sie Ihr vorhandenes Können auf diesem Gebiet anhand der Beispiele oder Übungsaufgaben. Je nach Erfolg dieser Tests können Sie sofort zum 2. Kapitel übergehen oder Ihr Wissen über das manuelle Rechnen auffrischen bzw. neu erwerben.

1.1. Die 4 Grundrechenarten

Wie die 4 Grundrechenarten Addition (+), Subtraktion (−), Multiplikation (×) und Division (÷) mit einem Taschenrechner durchgeführt werden, zeigen die folgenden Beispiele. Bei der Zahleneingabe ist statt des Kommas in einer Dezimalzahl der Punkt zu benutzen, und bei einer Zahl zwischen 0 und 1 darf die 0 vor dem Punkt weggelassen werden. Für 0,674 z.B. können wir 0.674 oder .674 eingeben.

Summe: 2,31 + 4,26

Eingabe/Taste	Anzeige
2.31	*2.31*
[+]	*2.31*
4.26	*4.26*
[=]	*6.57*

Differenz: 38,4 − 21,5

Eingabe/Taste	Anzeige
38.4	*38.4*
[−]	*38.4*
21.5	*21.5*
[=]	*16.9*

Produkt: 0,843 · 12,5

Eingabe/Taste	Anzeige
.843	*0.843*
[×]	*0.843*
12.5	*12.5*
[=]	*10.5375*

Quotient: $\frac{346{,}2}{51{,}8}$

Eingabe/Taste	Anzeige
346.2	*346.2*
[÷]	*346.2*
51.8	*51.8*
[=]	*6.683397683*

Bemerkungen: Eingetastete Zahlen werden in das *Anzeigeregister* (abgekürzt: AR oder X) gebracht und gleichzeitig im Sichtfenster des Rechners angezeigt. Für den Inhalt oder die Zahl des AR schreiben wir: (AR) = (X) = x. Betätigen wir nach der Zahleingabe eine Operationstaste [+], [−], [×] oder [÷], so wird die Zahl x in ein *Rechen-* oder *Ver-*

arbeitungsregister Y gebracht, d.h. es wird y = (Y) = x = (AR). Bringen wir durch Eintasten eine weitere Zahl in das AR, so wird nach Betätigen der Ergebnistaste *oder* einer weiteren Operationstaste die jeweilige arithmetische Rechenoperation ausgeführt.

Beispiel: 7 − 6 + 5 − 4

Eingabe/Taste	Anzeige X	Y
7	*7*	*0*
[−]	*7.*	*7*
6	*6*	*7*
[+]	*1.*	*7*
5	*5*	*1*
[−]	*6.*	*1*
4	*4*	*6*
[=]	*2.*	*6*

Für die Differenz 38,4 − 21,5 können wir auch schreiben: 38,4 + (− 21,5). In diesem Fall wird zu 38,4 die negative Zahl − 21,5 addiert.

Eine negative Zahl wird eingegeben, indem nach der Eingabe des Betrages der Zahl die Taste [+/−] betätigt wird.

Die Tastenfolge für die Berechnung von 38,4 + (− 21,5) sieht dann so aus: 38.4 [+] 21.5 [+/−] [=] mit dem Ergebnis 16.9. Dagegen würde 38.4 [+] [+/−] 21.5 [=] den falschen Wert 59.9 ergeben.

Für das Löschen von Zahlen gilt:

Die Taste [CLR] (Clear) löscht die Inhalte aller Rechenregister,
[CE] (Clear entry) löscht nur die unmittelbar eingetastete Zahl im AR.

Zahlendarstellungen

Berechnen wir mit dem Rechner das Produkt 8,16 · 2,435 = 19,8696, so hat sich der Punkt bei dem Ergebnis seinen Platz in der Anzeige selbst gesucht. Wir nennen dieses die *Fließpunktdarstellung.* Bei vielen Aufgaben wollen wir die Ergebnisse gar nicht auf so viele Nachkommastellen genau haben, wie der Rechner uns sie ausweist. So ist es z.B. ausreichend, in der Finanzmathematik die Ergebnisse auf 2 Nachkommastellen (auf Pfennige genau, wenn in DM gerechnet wird) und in vielen technischen Rechnungen die Längenangaben auf 3 Nachkommastellen (auf mm genau, wenn die Eingaben in m erfolgen) zu kennen. Werden Zahlen innerhalb einer Rechnung stets auf eine Anzahl von Nachkommastellen ausgegeben, so sprechen wir von einer *Festpunktdarstellung.*

Mit [*Fix] n werden die Rechenergebnisse mit fester Punktdarstellung auf n Nachkommastellen angegeben.

Wir werden im folgenden stets * für die Präfixtaste [2nd] schreiben, also z.B. [*Fix] statt ausführlich [2nd] [Fix (] .

Die nebenstehende Tabelle zeigt eine Rechnung mit 2 Nachkommastellen. Intern führt der Rechner alle Operationen mit der größeren Stellenzahl (der TI-58/59 bis zu 12 Stellen) aus. Wir können die Festpunktdarstellung durch [INV] (invers) [*Fix] aufheben. Dann werden alle Ziffern hinter dem Komma, die berechnet wurden, wieder sichtbar.

Eingabe/Taste	Anzeige
[*Fix]	*0.*
2	*0.00*
3.4725	*3.4725*
[×]	*3.47*
12.106	*12.106*
[=]	*42.04*
[INV]	*42.04*
[*Fix]	*42.038085*

Der Rechner besitzt in der Fließpunktdarstellung eine 10-ziffrige Anzeige, d.h. wir können in der bisherigen Form unmittelbar nur Zahlen von $.0000000001 = 10^{-10}$ bis $9999999999 = 9.999999999 \cdot 10^{9}$ eingeben. Kleinere oder größere Zahlen werden in der *Exponentialform* $m \cdot 10^{b}$ mit Hilfe der Taste [EE] (Enter Exponent) eingegeben.

Eingabe von $23{,}4 \cdot 10^{-16}$	
Eingabe/Taste	Anzeige
23.4	*23.4*
[EE]	*23.4 00*
16	*23.4 16*
[+/−]	*23.4−16*

Eingabe von $0{,}752 \cdot 10^{34}$	
Eingabe/Taste	Anzeige
.752	*0.752*
[EE]	*0.752 00*
34	*0.752 34*

Der Rechner schaltet selbständig auf die Exponentialform um, wenn sich bei Rechenoperationen Zahlen ergeben, die kleiner als 10^{-10} oder größer als $9{,}999999999 \cdot 10^{9} \approx 10^{10}$ sind.

Der Rechner gibt übrigens beim Rechnen in der Exponentialdarstellung (man nennt sie auch: scientific notation) die Ergebnisse stets in der normierten Form mit $1 \leq m < 10$ an. Wir nennen dieses die *Gleitpunktdarstellung* einer Zahl mit der Mantisse m und dem Exponenten b.

Beispiel: $83{,}48 \cdot 10^{5} + 6{,}01 \cdot 10^{6} =$ [*1.4358 07*] $= 1{,}4358 \cdot 10^{7}$

1.2. Termberechnungen

Die TI-Taschenrechner rechnen nach dem Algebraischen Operations-System (AOS), einem Logiksystem mit Hierarchien[1]:

> Zahlen und Operationen werden eingegeben wie (von links nach rechts) geschrieben. Multiplikation und Division haben Vorrang vor Addition und Subtraktion (kurz: Punktrechnung geht vor Strichrechnung), wenn nicht durch Klammern eine andere Rangfolge festgelegt wird.

[1] Über Logiksysteme informiert ausführlich: H.SCHUMNY: Taschenrechner Handbuch, Vieweg 1976.

Die folgenden Beispiele sollen dem im manuellen Rechnen weniger geübten Leser zur Erläuterung dienen. Bei vielen Aufgaben führen unterschiedliche Tastenfolgen zum Ziel. Versuchen Sie selbst, weitere Varianten herauszufinden. Die Benutzung der Tasten [x²], [1/x], [√x] oder [•|x|] dürfte keine Schwierigkeiten bereiten. Nach dem Betätigen einer dieser Tasten wird mit dem im AR stehenden Zahlenwert x die auf der Taste stehende Rechenoperation ausgeführt und der neue Zahlenwert x^2, $1/x$, $\sqrt{x}$ oder $|x|$ im AR angezeigt.

Übungen:

Aufgabe	Tastenfolge	Ergebnis
$3{,}4 + 2 \cdot 1{,}82$	3.4 [+] 2 [x] 1.82 [=]	*7.04*
$(3{,}4 + 2) \cdot 1{,}82$	[(] 3.4 [+] 2 [)] [x] 1.82 [=] oder: 3.4 [+] 2 [=] [x] 1.82 [=]	*9.828*
$\frac{6{,}5 + 3 \cdot 1{,}4^2}{2{,}6}$	6.5 [+] 3 [x] 1.4 [x²] [=] [÷] 2.6 [=]	*4.761538462*
$\frac{24{,}8}{4 \cdot 5{,}2^2 - \frac{1}{0{,}074}}$	24.8 [÷] [(] 4 [x] 5.2 [x²] [−] .074 [1/x] [)] [=]	*.2620276877*
$\frac{468 - 305}{2 \cdot (104 + 217)}$	468 [−] 305 [=] [÷] 2 [÷] [(] 104 [+] 217 [)] [=]	*0.253894081*
$\frac{0{,}65 + \frac{0{,}86}{\sqrt{1{,}47 - 0{,}92}}}{0{,}0584}$	.65 [+] .86 [÷] [(] 1.47 [−] .92 [)] [√x] [=] [÷] .0584 [=]	*30.98670828*
$\frac{43{,}7 - \frac{123{,}8}{8{,}04 + 2 \cdot 6{,}13}}{\left(18{,}3 + \frac{40{,}7}{3{,}6}\right)^2}$	43.7 [−] 123.8 [÷] [(] 8.04 [+] 2 [x] 6.13 [)] [=] [÷] [(] 18.3 [+] 40.7 [÷] 3.6 [)] [x²] [=]	*.0429001177*
$\cfrac{2}{1 + \cfrac{3}{4 - \cfrac{6}{5 + \frac{7}{8}}}}$	2 [÷] [(] 1 [+] 3 [÷] [(] 4 [−] 6 [÷] [(] 5 [+] 7 [÷] 8 [)] [)] [)] [=]	*.9964412811*

Aufgabe	Tastenfolge	Ergebnis
$\frac{314{,}6 \cdot 10^{14} - 2{,}47 \cdot 10^{16}}{0{,}745 \cdot 10^{8} \cdot 1{,}602 \cdot 10^{-5}}$	314.6 EE 14 − 2.47 EE 16 = ÷ .745 EE 8 ÷ 1.602 EE 5 +/− =	*5.6640609 12*
$\left(\frac{5800347}{0{,}000156} + 672 \cdot 10^{8}\right) \frac{0{,}0706}{981050}$	5800347 ÷ .000156 + 672 EE 8 = x .0706 ÷ 981050 = INV EE	*7.5116955 03* *7511.695464*

Sehen wir uns das drittletzte Beispiel noch einmal etwas genauer an. Durch Betätigen der Taste) wird jeweils eine unvollständige Operation abgeschlossen:

$$5 + \frac{7}{8} = 5{,}875; \quad 4 - \frac{6}{5{,}875} = 2{,}978723404; \quad 1 + \frac{3}{2{,}978723404} = 2{,}007142857\,.$$

Hätten wir nach der 8 unmittelbar die Taste = gedrückt, so würde der Rechner alle unvollständigen Operationen ‚aufarbeiten' und sofort den Endwert 0,9964412811 ausgeben. Überprüfen Sie diese Art des Abschlusses selbst mit Ihrem Rechner.

1.3. Datenspeicher (Datenregister, Memories)

Bei der Berechnung der Zahl

$$z = 4\left(0{,}5 + \frac{3\pi}{4}\right) + \frac{3}{5 + \frac{1}{2}\left(0{,}5 + \frac{3\pi}{4}\right)} - 2\left(0{,}5 + \frac{3\pi}{4}\right)^2 + \frac{7}{0{,}5 + \frac{3\pi}{4}}$$

werden wir beachten, daß der Term $0{,}5 + \frac{3\pi}{4}$ an 4 verschiedenen Positionen steht. Es erscheint sinnvoll, die Berechnung aufzuteilen in

$$y = 0{,}5 + \frac{3\pi}{4} \quad \text{und} \quad z = 4y + \frac{3}{5 + \frac{1}{2}y} - 2y^2 + \frac{7}{y}\,.$$

Den Zahlenwert y können wir im Rechner in einem seiner **Datenspeicher** (Memories) aufbewahren und von dort jederzeit wieder zur Weiterverarbeitung in das AR zurückholen. Nach dem Einschalten des TI-58 (59) stehen dem Benutzer 30 (60) Datenspeicher[1] zur Verfügung, die wir mit R_{00}, R_{01}, R_{02} usw. (kurz: R_{nm}) bezeichnen.
Wir führen folgende Abkürzungen ein:

(R_{nm}) : Inhalt des Datenspeichers R_{nm}; also die in R_{nm} gespeicherte Zahl.
$(AR) \rightarrow R_{nm}$: Der Inhalt des AR wird in den Speicher R_{nm} gebracht.

Die Speichertasten STO (Store), RCL (Recall), *Exc (Exchange) und *CM$_s$ (Clear all Memories) besitzen die folgende Bedeutung:

- STO n m: $(AR) \rightarrow R_{nm} \wedge (AR)$ bleibt im AR erhalten,
- RCL n m: $(R_{nm}) \rightarrow AR \wedge (R_{nm})$ bleibt in R_{nm} erhalten,
- *Exc n m: $(R_{nm}) \rightarrow AR \wedge (AR) \rightarrow R_{nm}$; kurz: $(R_{nm}) \leftrightarrow (AR)$,
- *CM$_s$: löscht die Inhalte aller R_{nm}, (AR) bleibt erhalten.

[1] Wer mehr Speicherplätze benötigt, findet Näheres hierüber in 2.2.

Bemerkungen:

1. Die Speichertasten dürfen in eine *unvollständige* Operation gesetzt werden. Zum Beispiel wird in der Tastenfolge 2 [STO] 15 [+] 3 [=] die Zahl 2 nach R_{15} gespeichert und danach die Zahl 3 zu der im AR stehenden 2 addiert. Dieselbe Wirkung würde durch 2 [+] [STO] 15 3 [=] erreicht.

2. Durch [•CM$_s$] werden die Inhalte *aller* R_{nm} durch 0 ersetzt. Soll nur der Inhalt *eines* Speichers, z.B. R_{26}, gelöscht werden, so erreichen wir dieses mit den Tasten 0 [STO] 26.

3. Für die 10 Datenspeicher R_{00} bis R_{09} schreiben wir auch R_n ($n \in \mathbb{N}_{0,9}$). Für diese Datenspeicher gilt die **Kurzformadressierung**: Für (AR) → R_n kann immer dann [STO] n geschrieben werden, wenn unmittelbar nach der einziffrigen Adresse n keine Zifferntaste folgt. Entsprechendes gilt für [RCL] n oder [•Exc] n. Wir dürfen z.B. 2 [STO] 5 [+] 3 [=] statt [STO] 05 eingeben. Dagegen ist 2 [+] [STO] 5 3 [=] nicht zulässig, hier folgt auf die Speicheradresse 5 die Ziffer 3. Wollen wir aber (R_n) im AR ablesen, so benutzen wir [RCL] on.

Beispiel: Berechnung der obigen Zahl z. Der Zwischenwert y soll nach R_0 gespeichert werden: y → R_0.

Tastenfolge: . 5 [+] 3 [×] [•π] [÷] 4 [=] [STO] 0 [×] 4 [+] 3 [÷] [(] 5 [+]
[RCL] 0 [÷] 2 [)] [−] 2 [×] [RCL] 0 [x²] [+] 7 [÷] [RCL] 0 [=]

Ergebnis: z = − 1,973401385.

Beispiel: Für einen geraden Kreiszylinder mit dem Durchmesser d = 24,3 cm und der Höhe h = 18,6 cm sollen berechnet werden:

die Mantelfläche $A_M = \pi d h$;

die Oberfläche $A = A_M + \frac{\pi}{2} d^2$;

das Volumen $V = \frac{\pi}{4} d^2 h = \frac{1}{4} A_M d$.

Die Ergebnisse sollen in cm² bzw. cm³ auf eine Nachkommastelle genau angegeben werden.

Mit der Speicherfestlegung d → R_1 und A_M → R_2 sieht die Berechnung folgendermaßen aus:

Eingabe/Taste	Anzeige	Bemerkungen
[•Fix]	*0.*	(AR) auf eine Nachkommastelle
1	*0.0*	
[•π]	*3.1*	π
[×]	*3.1*	
24.3	*24.3*	Eingabe d
[STO]	*24.3*	d → R_1
1	*24.3*	
[×]	*76.3*	π · d
18.6	*18.6*	Eingabe h
[=]	*1419.9*	Ergebnis A_M
[STO]	*1419.9*	A_M → R_2
2	*1419.9*	
[+]	*1419.9*	A_M
[•π]	*3.1*	
[÷]	*3.1*	
2	*2*	

Eingabe/Taste	Anzeige	Bemerkungen
[x]	*1.6*	$\frac{\pi}{2}$
[RCL]	*1.6*	$(R_1) = d \rightarrow AR$
01	*24.3*	
[x^2]	*590.5*	d^2
[=]	*2347.5*	Ergebnis A
[RCL]	*2347.5*	$(R_2) = A_M \rightarrow AR$
02	*1419.9*	
[÷]	*1419.9*	A_M
4	*4*	
[x]	*355.0*	$A_M/4$
[RCL]	*355.0*	$(R_1) = d \rightarrow AR$
01	*24.3*	
[=]	*8626.1*	Ergebnis V

Ergebnis: $A_M = 1419{,}9\ cm^2$; $A = 2347{,}5\ cm^2$; $V = 8626{,}1\ cm^3$.

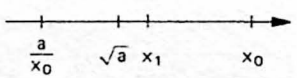
Beispiel: Es soll die Quadratwurzel $x = \sqrt{a}$ (ohne Benutzung der Taste [$\sqrt{x}$]) berechnet werden.

Ist x_0 ein Näherungswert für x, so berechnen wir einen 2. Näherungswert x_1 nach der Vorschrift:

$$x_1 = \frac{1}{2}\left(x_0 + \frac{a}{x_0}\right).$$

Erläuterung des Verfahrens: Ist $x_0 > \sqrt{a}$, so ist $\frac{a}{x_0} < \sqrt{a}$.

Der arithmetische Mittelwert x_1 aus diesen Zahlen x_0 und $\frac{a}{x_0}$ (s. Darstellung auf der Zahlengeraden) wird daher im allgemeinen eine bessere Näherung für $\sqrt{a}$ sein als x_0. Entsprechendes gilt, wenn $x_0 < \sqrt{a}$ und damit $\frac{a}{x_0} > \sqrt{a}$ ist.



Den errechneten Wert x_1 machen wir zum neuen Ausgangswert x_0 und berechnen nach obiger Vorschrift den nächsten Näherungswert, d.h. wir wiederholen (iterieren) denselben Rechnungsgang mit einem neuen Zahlenwert. Rechenmethoden dieser Art, die in der numerischen Mathematik häufig benutzt werden, nennt man **Iterationsmethoden** oder **Iterationsverfahren.** Die Folge $x_1, x_2, x_3 \ldots$, die wir nach der obigen Iterationsvorschrift erhalten, konvergiert gegen den Grenzwert $x = \sqrt{a}$ [1].

[1] Beweise über Konvergenz von Folgen findet der Leser in den Lehrbüchern der Analysis.

Wir führen die Rechnung durch für $a = 6{,}5$ mit $x_0 = 2$ und iterieren so oft, bis eine gewünschte Genauigkeit erreicht ist.

Speicherplan: $a = 6{,}5 \rightarrow R_1$, $x_0 \rightarrow R_2$.

Tastenfolge: 6.5 [STO] 01 2 [STO] 2 [+] [RCL] 1 [÷] [RCL] 2 [=] [÷] 2 [=]

Ergebnis: $x_1 = 2{,}625$.

Bei der Berechnung der nächsten x-Werte brauchen wir die Speicherung von $a = 6{,}5$ nach R_1 nicht erneut vorzunehmen. Auch die Eingabe des nächsten Ausgangswertes (x_0) ist nicht erforderlich, dieser Wert steht bereits als Ergebnis der vorhergehenden Rechnung im AR. Mit der Tastenfolge

[STO] 2 [+] [RCL] 1 [÷] [RCL] 2 [=] [÷] 2 [=]

erhalten wir insgesamt die Zahlenfolge

2; 2,625; 2,55059238; 2,549509988; 2,549509757; 2,549509757 .

Ergebnis: $\sqrt{6{,}5} = 2{,}549509757$.

Für das Arbeiten mit Speichern bieten die folgenden Tasten oftmals Rechenerleichterungen:

	[SUM]	n m:	$(R_{nm}) + (AR) \rightarrow R_{nm}$, d.h. der Inhalt des AR wird zum Inhalt des Speichers R_{nm} addiert und in R_{nm} gespeichert. (AR) bleibt im AR erhalten.
[INV]	[SUM]	n m:	$(R_{nm}) - (AR) \rightarrow R_{nm}$.
	[•Prd]	n m:	$(R_{nm}) \cdot (AR) \rightarrow R_{nm}$.
[INV]	[•Prd]	n m:	$\frac{(R_{nm})}{(AR)} \rightarrow R_{nm}$.

Bemerkung: Auch für diese Speichertasten gilt die oben beschriebene Kurzformadressierung.

Beispiel: Für $x_1 = 4$, $x_2 = 5$, $x_3 = 6$ sollen $s_1 = x_1 + x_2 + x_3$, $s_2 = 2x_1 + 3x_2 + 4x_3$, $p = x_1 x_2 x_3$, $a = s_1 s_2$ und $b = \frac{1}{p} + \frac{s_1}{s_2}$ berechnet werden.

Speicherplan: $s_1 \rightarrow R_1$, $s_2 \rightarrow R_2$, $p \rightarrow R_3$.

Eingabe/Taste	Anzeige	Bemerkungen
[•CM$_s$]	*0.*	Löschen aller R_n
4	*4*	Eingabe $x_1 = 4$
[SUM]	*4.*	$0 + 4 = 4 \rightarrow R_1$
1	*4.*	
[STO]	*4.*	$4 \rightarrow R_3$
3	*4.*	
[×]	*4.*	
2	*2*	
[=]	*8.*	$2 \cdot 4 = 8$

Eingabe/Taste	Anzeige	Bemerkungen
SUM	8.	$0 + 8 = 8 \rightarrow R_2$
02	8.	
5	5	Eingabe $x_2 = 5$
SUM	5.	$4 + 5 = 9 \rightarrow R_1$
1	5.	
*Prd	5.	$4 \cdot 5 = 20 \rightarrow R_3$
3	5.	
×	5.	
3	3	
=	15.	$3 \cdot 5 = 15$
SUM	15.	$8 + 15 = 23 \rightarrow R_2$
02	15.	
6	6	Eingabe $x_3 = 6$
SUM	6.	$9 + 6 = 15 \rightarrow R_1$
1	6.	
*Prd	6.	$20 \cdot 6 = 120 \rightarrow R_3$
3	6.	
×	6.	
4	4	
=	24.	$4 \cdot 6 = 24$
SUM	24.	$23 + 24 = 47 \rightarrow R_2$
2	24.	
RCL	24.	$(R_1) = s_1 = 15 \rightarrow AR$
01	15.	
×	15.	
RCL	15.	$(R_2) = s_2 = 47 \rightarrow AR$
02	47.	
=	705.	$s_1 \cdot s_2 = a$
RCL	705.	$(R_3) = p = 120 \rightarrow AR$
03	120.	
1/x	.0083333333	Reziprokwert von p
+	.0083333333	
RCL	.0083333333	$s_1 \rightarrow AR$
01	15.	

Eingabe/Taste	Anzeige	Bemerkungen
[÷]	*15.*	
[RCL]	*15.*	$s_2 \rightarrow AR$
02	*47.*	
[=]	*.3274822695*	$\frac{1}{p} + \frac{s_1}{s_2} = b$

Ergebnis: $s_1 = 15$; $s_2 = 47$; $p = 120$; $a = 705$; $b = 0{,}3274822695$.

1.4. Funktionstasten

Den Gebrauch der verschiedenen Funktionstasten wollen wir an einigen Beispielen erläutern.

[y^x] und [INV] [y^x]

Zulässig für $y \geq 0$. Mit [INV] [y^x] wird $\sqrt[x]{y} = y^{1/x}$ berechnet. Für $y < 0$ wird der Wert $|y|^x$ oder $\sqrt[x]{|y|}$ durch Blinken angezeigt.

Einfache Berechnungen:

Berechnet werden soll	Eingabe	Taste	Eingabe	Taste	Ergebnis
2^3	2	[y^x]	3	[=]	*8*
$\sqrt[5]{32}$	32	[INV] [y^x]	5	[=]	*2*
$32^{1/5} = 32^{0,2}$	32	[y^x]	.2	[=]	*2*
$2{,}84^{-0{,}74}$	2.84	[y^x]	−.74	[=]	*0,4618962415*
$\sqrt[3,8]{0{,}0614}$	.0614	[INV] [y^x]	3.8	[=]	*0,4798405429*

Zur Berechnung von $\sqrt[x]{y} = y^{1/x}$ kann selbstverständlich auch die Tastenfolge y [y^x] x [1/x] = benutzt werden.

Für *zusammengesetzte Berechnungen* gilt die Ergänzung zur Hierarchieregel (s. 1.1.):

> Die Berechnung von Potenzen y^x oder $\sqrt[x]{y}$ hat Vorrang vor den Grundrechenarten (kurz: Potenzrechnung geht vor Punktrechnung, diese vor Strichrechnung).

Zum Beispiel wird $1 + 4 \cdot 2^3$ mit der

Tastenfolge: 1 [+] 4 [x] 2 [y^x] 3 [=]

oder: 2 [y^x] 3 [x] 4 [+] 1 [=]

berechnet. Dagegen wird mit

2 [y^x] [(] 3 [x] 4 [+] 1 [)] [=]

der Zahlenwert $2^{3 \cdot 4 + 1} = 2^{13} = 8192$ berechnet.

Oder mit

[(] 1 [+] 4 [×] 2 [)] [y^x] 3 [=]

der Zahlenwert $9^3 = 729$.

Bei den folgenden Beispielen sollten Sie sich vor jedem Betätigen einer Taste genau überlegen, was der Rechner danach ausführen wird.

Beispiel: $\dfrac{3 \cdot 6{,}47^{0{,}84} + 1{,}31^{\frac{42}{29}}}{\sqrt[5]{76{,}5}}$

Tastenfolge: 3 [×] 6.47 [y^x] .84 [+] 1.31 [y^x] [(] 42 [÷] 29 [)] [=] [÷] 76.5 [INV] [y^x] 5 [=]

Ergebnis: 6,668144411.

Beispiel: $\dfrac{\sqrt[7]{1 + 1{,}6^2} - \left(\frac{0{,}86}{1{,}21}\right)^{-2{,}07}}{6 \cdot 0{,}98^{0{,}75}}$

Tastenfolge: 1 [+] 1.6 [x^2] [=] [y^x] 7 [1/x] 7 [−] [(] .86 [÷] 1.21 [)] [y^x] 2.07 [+/−] [=] [÷] 6 [÷] .98 [y^x] .75 [=]

Ergebnis: −0,1402049962.

[ln x] [•log]

$\log x = \lg x = \log_{10} x$ und $\ln x = \log_e x$ sind definiert für $x > 0$. Für $x < 0$ wird $\log|x|$ oder $\ln|x|$ berechnet und durch Blinken angezeigt. e^x wird mit [INV] [ln x], 10^x mit [INV] [•log] berechnet.

Beispiel: $\dfrac{\ln(3 + \sqrt{5}) \cdot 10^{0{,}72}}{e^{-\frac{3{,}28}{2{,}64}}}$

Tastenfolge: 3 [+] 5 [$\sqrt{x}$] [=] [ln x] [×] .72 [INV] [•log] [÷] [(] 3.28 [÷] 2.64 [)] [+/−] [INV] [ln x] [=]

Ergebnis: 30,09717675.

Beispiel: $\dfrac{\sqrt{(\ln 1{,}84)^2 + 3}}{4 \cdot \ln(1{,}3 + e^{1{,}5})^2}$

Tastenfolge: 1.84 [ln x] [x^2] [+] 3 [=] [$\sqrt{x}$] [÷] 4 [÷] [(] 1.3 [+] 1.5 [INV] [ln x] [)] [x^2] [ln x] [=]

Ergebnis: 0,1308097056.

[•sin] [•cos] [•tan]

Für den im AR angezeigten Winkel erhalten wir den Wert der jeweiligen trigonometrischen Funktion.

Einen Winkel können wir in einem der drei Winkelmaße eingeben:

Gradmaß (Taste [•Deg] (Degree) betätigen),
neues Gradmaß (Taste [•Grad] (Gon) betätigen),
Bogenmaß (Taste [•Rad] (Radiant) betätigen).

Wird der Rechner eingeschaltet, so wird der Winkel immer im (alten) Gradmaß gelesen.

Es gilt: $360° = 400^g = 2\pi$ (rad).

Es ist vielfach üblich, bei der Angabe eines Winkels im Bogenmaß rad wegzulassen, also z.B. $180° = \pi$ zu schreiben. Wir werden in diesem Buch ebenfalls den Winkel im Bogenmaß ohne rad angeben.

Ermittlung der Hauptwerte der Arcusfunktionen:

zulässige Variable x = (AR)	Funktion y	Tasten	Gradmaß [•Deg]	neues Gradmaß [•Grad]	Bogenmaß [•Rad]
$\|x\| \leqq 1$	arc sin x	[INV] [•sin]	$-90° \leqq y \leqq 90°$	$-100^g \leqq y \leqq 100^g$	$-\frac{\pi}{2} \leqq y \leqq \frac{\pi}{2}$
$\|x\| \leqq 1$	arc cos x	[INV] [•cos]	$0° \leqq y \leqq 180°$	$0^g \leqq y \leqq 200^g$	$0 \leqq y \leqq \pi$
$\|x\| < 10^{10}$	arc tan x	[INV] [•tan]	$-90° (\leqq) y (\leqq) 90°$	$-100^g (\leqq) y (\leqq) 100^g$	$-\frac{\pi}{2} (\leqq) y (\leqq) \frac{\pi}{2}$

Die häufiger auftretenden Umrechnungen vom Grad- auf das Bogenmaß und umgekehrt sind in der folgenden Tabelle zusammengestellt. (Auf den Hauptwert achten!)

Im (AR) steht der Winkel im	Tastenfolge	Im (AR) erscheint der Winkel im
Gradmaß	[×] [•π] [÷] 180 [=] oder: [•sin] [•Rad] [INV] [•sin]	Bogenmaß
Bogenmaß	[×] [180] [÷] [•π] [=] oder: [•sin] [•Deg] [INV] [•sin]	Gradmaß

Beispiel: $\dfrac{\sin 25{,}6° + 2 \cdot \cos \frac{81{,}6°}{1{,}2}}{\sqrt{1 + 3 \cdot \tan^2 18{,}7°}}$

Tastenfolge: [•Deg] 25.6 [•sin] [+] 2 [×] [(] 81.6 [÷] 1.2 [)] [•cos] [=] [÷]
[(] 1 [÷] 3 [×] 18.7 [•tan] [x^2] [)] [$\sqrt{x}$] [=]

Ergebnis: 1,019077536.

Beispiel: $4 \cdot \text{arc sin} \dfrac{\sqrt{1 + 0{,}25^2}}{2} - \text{arc tan} \dfrac{5}{12}$ (Berechnung in Grad- und Bogenmaß auf 2 Nachkommastellen).

Tastenfolge: [•Deg] [•Fix] 2 1 [+] .25 [x^2] [=] [$\sqrt{x}$] [÷] 2 [=] [INV] [•sin]
[×] 4 [−] [(] 5 [÷] 12 [)] [INV] [•tan] [=] [×] [•π] [÷] 180 [=]

Ergebnis: $101{,}47° = 1{,}77$.

1.5. Übungsaufgaben

Notieren Sie die vollständige Tastenfolge für die jeweilige Aufgabe, bevor Sie mit dem Eintasten in den Rechner beginnen.

1.1. Berechnen Sie

a) $\dfrac{768 + 3 \cdot \sqrt{1025}}{2 \cdot (7{,}8 + 5 \cdot 3{,}1)^2}$

b) $\dfrac{25{,}8 - \frac{18{,}7}{1{,}02 - 0{,}34}}{(2{,}1 + 6{,}8) \cdot \sqrt{0{,}746 + 0{,}132}}$

c) $\dfrac{6 \cdot 1{,}34^{1{,}72} + 5 \cdot \sqrt[4]{20{,}63}}{0{,}65^{7{,}21 + 5{,}43}}$

d) $\dfrac{4 \cdot \sqrt[5]{2{,}38 + 1{,}41^{3{,}68}} + 1}{(4{,}08^2 \cdot 0{,}81 + \sqrt{18{,}3})^2 + 2}$

e) $10^{14{,}9} \cdot \ln \dfrac{0{,}41}{8023}$

f) $\dfrac{\ln(1 + \sin 4{,}68^\circ)}{\cos\left(1{,}42 \cdot \ln \frac{2}{1{,}65}\right)}$

g) $\ln(14{,}2 - 2{,}5 \cdot 8{,}4^{1/3}) + 2 \cdot e^{-\sqrt{0{,}876}}$

h) $\frac{1}{2}\left(\arcsin \dfrac{4}{4{,}3 + \sqrt{12{,}6}} + 3 \cdot \arccos e^{-\frac{268}{308}}\right)$

1.2. Welchen Zahlenwert zeigt der Rechner in den Tastenfolgen a), b) und c) nach Betätigen der letzten [=]-Taste an? Überprüfen Sie das von Ihnen vorausgesagte Ergebnis mit dem Rechner.

a) 2 [+] 3 [×] 16 [√x] [−] 8 [÷] 2 [=] [÷] [(] 4 [x²] [+] 3 [x²] [)] [√x] [=]

b) [(] 4 [+] 2 [)] [÷] 3 [=] [STO] 07 13 [−] [RCL] 7 [yˣ] 3 [=] [STO] 8 [−] 6 [+] [RCL] 7 [×] [RCL] 8 [=]

c) 4 [×] 20 [+] 1 [=] [INV] [yˣ] 4 [yˣ] 2 [+/−] [×] 18 [+] 1 [=] [yˣ] 1 [+/−] [+] 2 [÷] 3 [=]

1.3. Berechnen Sie für $z_1 = 8$, $z_2 = 4$, $z_3 = 2$

$s_1 = z_1 - z_2 + z_3$; $s_2 = z_1^2 + z_2^2 + z_3^2$; $p = z_1^2 z_2^2 z_3^2$; $q = (z_1/z_2)/z_3$; $r = s_1 s_2 q \sqrt[6]{p}$.

Tasten Sie bei dieser Aufgabe jeden der z-Werte nur einmal ein.

1.4. Berechnen Sie für einen geraden Kreiskegel mit dem Durchmesser $d = 18{,}4$ dm und der Höhe $h = 12{,}3$ dm

die Mantelfläche $A_M = \frac{\pi}{2} ds = \frac{\pi}{2} d \sqrt{\left(\frac{d}{2}\right)^2 + h^2}$;

die Oberfläche $A = \frac{\pi}{4} d^2 + A_M$;

das Volumen $V = \frac{\pi}{12} d^2 h$.

Geben Sie A_M und A in dm^2 und V in dm^3 auf eine Nachkommastelle an.

2. Programmaufbau und Programmherstellung

2.1. Ein einfaches Programm entsteht

Für ein Quadrat mit zwei angesetzten Halbkreisen (Bild 2.1.1) berechnen wir den Flächeninhalt

$$A = \left(1 + \frac{\pi}{4}\right) d^2$$

für d = 2 cm mit der Tastenfolge

[(] 1 [+] [•π] [÷] 4 [)] [×] 2 [x²] [=]

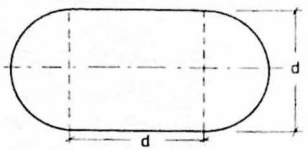


Bild 2.1.1

Soll der Flächeninhalt A für mehrere (z.B. für 8) verschiedene Durchmesser d ausgerechnet werden, so könnten wir jedesmal die obigen Tasten mit dem jeweiligen Wert von d betätigen. Wir hätten dann (abgesehen von der Eingabe von d) 8 · 10 = 80 Tasten in immer wieder derselben Reihenfolge zu drücken. Diese Arbeit der achtmaligen (oder noch öfteren) Wiederholung nimmt uns der programmierbare Rechner ab. Wir können dabei die obige Tastenfolge bis auf wenige zusätzliche ‚Anweisungen' bereits als ‚Programm' benutzen, das wir dem Rechner nur *einmal* einzugeben brauchen, um dann mit wenigen Tasten für einen Wert von d den Flächeninhalt A auszurechnen.

Unter einer **Anweisung** (Befehl) wollen wir jede Vorschrift verstehen, die wir dem Rechner durch Tastendruck mitteilen können. Zum Beispiel würde die Anweisung

[x²] bewirken: Quadriere den im AR stehenden Zahlenwert,

oder [RCL] 13: Bringe den im Speicher R_{13} stehenden Zahlenwert in das Anzeigeregister, kurz: $(R_{13}) \rightarrow AR$,

oder [•CM$_s$] : Lösche alle Datenspeicher.

Eine Folge von Anweisungen bezeichnen wir als Programm.

Daß dabei diese Folge auch eine sinnvolle Folge sein soll, versteht sich eigentlich von selbst. So würde z.B.

[RCL] [+] 3 [•Fix] 4 [÷] [=]

keine sinnvolle (wir würden sagen: keine zulässige) Tastenfolge sein. Probieren Sie selbst, was Ihnen der Rechner als Antwort auf diese Tastenfolge gibt! Überlegen Sie, was nicht zulässig ist.

Der wesentliche Unterschied zwischen einem nicht-programmierbaren und einem programmierbaren Rechner besteht darin, daß dieser nicht nur Daten (Zahlen) speichern kann, wie wir in 1.3. gesehen haben, sondern auch Anweisungen wie [STO] 3, [+], [SUM] 2 5 usw. Der programmierbare Rechner führt diese gespeicherten Anweisungen später selbständig in der Reihenfolge der eingegebenen Speicherung aus.

Geben Sie für unser obiges Beispiel dem Rechner diese Tastenfolge ein:

[LRN] [(] 1 [+] [•π] [÷] 4 [)] [×] [R/S] [x²] [=] [R/S] [RST] [LRN] [RST]

Betätigen Sie jetzt [R/S] 2 [R/S], so erscheint im AR der Wert A = 7,141592654 für d = 2. Oder mit [R/S] 3 [R/S]: A = 16,06858347 für d = 3.

Wir haben ein einfaches Programm zur Berechnung des Flächeninhalts A für verschiedene Durchmesser d benutzt, ohne genau zu wissen, was die oben benutzten Tasten [LRN], [R/S], [RST] im einzelnen bewirken.

Bisher haben wir den Rechner immer benutzt, um unmittelbar mit ihm zu rechnen. Wir sagen auch: Der Rechner arbeitet in der Betriebsart **RECHNEN**. Daneben gibt es die Betriebsart **LEARN**, in der dem Rechner Programme eingegeben werden können. Die einzelnen Anweisungen werden in den Programmspeichern, die von 000, 001, 002 usw. durchnumeriert sind, in verschlüsselter Form in der Reihenfolge der Eingabe gespeichert.

In diese Betriebsart LEARN schalten wir den TI-58 oder TI-59 aus der Betriebsart RECHNEN mit der Taste [LRN]. Nach dem Einschalten und Betätigen dieser Taste erscheint im Sichtfenster des Rechners:

000 00

Die ersten drei Ziffern geben die **Programmspeicherstelle** (abgekürzt: PSS) oder **Befehlsadreßregisterstelle** (BARS) an, die letzten beiden Ziffern dienen zur numerischen Festlegung der einzelnen Tasten. So besitzt z.B. [RCL] als Taste in der 4. Zeile und 3. Spalte auf dem Tastenfeld des Rechners den Tastenkode 43 oder die Taste [*Exc] den Kode 48. Die Anzeige [*156 34*] z.B. würde bedeuten, daß in diesem Programm in der Befehlsadreßregisterstelle 156 die Anweisung [$\sqrt{x}$] (Tastenkode 34) gespeichert ist. Im Anhang (8.1.) finden Sie für die Tasten, die in diesem Buch benutzt werden, den Tastenkode angegeben.

Die oben angegebene Tastenfolge beginnt in der Betriebsart LEARN folgendermaßen:

PSS	Kode	Taste
000	53	(
001	01	1
002	85	+
003	89	*π
004	55	÷

usw.

Beim Aufschreiben eines Programms werden wir später die Durchnumerierung der Programmspeicherstellen und daneben die zu betätigende Taste (ab jetzt ohne Tastenumrandung) schreiben. Die Kenntnis der PSS oder BARS kann bei einigen Aufgaben von Wichtigkeit sein. Den Tastenkode werden wir später nicht mitschreiben; er wird nur zur Überprüfung eines gespeicherten Programms von Bedeutung sein (s. 2.3.).

Nach der Eingabe eines Programms schalten wir den Rechner aus der Betriebsart LEARN mit der Taste [LRN] zurück in die Betriebsart RECHNEN. Merken wir uns:

> Die Taste [LRN] schaltet den Rechner aus der Betriebsart RECHNEN in die Betriebsart LEARN und umgekehrt.

Das Befehlsadreßregister (BAR) wird beim Umschalten von einer in die andere Betriebsart an *der* Programmspeicherstelle stehenbleiben, die es in der Betriebsart LEARN zuletzt innehatte. Wird in unserem Beispiel an der PSS 004 die Taste [÷] bestätigt, so erscheint [*005 00*]. Diese Speicherstelle erscheint wieder nach zweimaligem Betätigen von [LRN].

Bei der Durchführung eines Programms – wir sprechen dann von der Betriebsart RUN – soll der Ablauf eines Programms häufig mit *der* Anweisung beginnen, die im Befehlsadreßregister in der Speicherstelle 000 steht.

> Mit [RST] (Reset) wird das Befehlsadreßregister auf die Speicherstelle 000 zurückgestellt. [RST] kann sowohl als Anweisung in einem Programm als auch manuell in der Betriebsart RECHNEN benutzt werden.

Die Taste [RST] werden wir ebenfalls betätigen, wenn wir zur Eingabe eines Programms mit [LRN] in die Betriebsart LEARN schalten und nicht ganz sicher sind, ob infolge vorhergehender Benutzung des Rechners das BAR bereits auf 000 steht. Bei einem gerade erst eingeschalteten Rechner ist natürlich diese Taste nicht erforderlich.

Fassen wir zusammen:

Eingabe des Programms:

1. Den Rechner mit den Tasten [RST] [LRN] auf die Programmspeicherstelle 000 in der Betriebsart LEARN schalten.
2. Vollständiges Programm eintasten.
3. Mit [LRN] in die Betriebsart RECHNEN schalten.
4. Mit [RST] das Befehlsadreßregister auf 000 stellen.

Bei den späteren Beispielen in diesem Buch wird vorausgesetzt, daß das Programm in dieser Weise eingegeben wird. Der Punkt 4. wird bei sehr vielen Aufgaben nicht erforderlich sein, s. hierzu z.B. den Abschnitt 2.4. über Programmtasten.

Die Anzahl der Programm- und Datenspeicher ist bei den TI-Rechnern variabel. Nach dem Einschalten stehen beim TI-58 (59) 240 (480) Programmspeicher und 30 (60) Datenspeicher zur Verfügung. Wir werden in diesem Buch mit dieser Kombination beim TI-58 stets auskommen. Wer mehr Programmspeicher benötigt, muß dafür auf Datenspeicher verzichten und umgekehrt. Beim TI-58 ist die Wahl dieser Speicher nach nebenstehender Tabelle möglich.

Progr.-Sp.	Datensp.
480	0
400	10
320	20
240	30
160	40
80	50
0	60

Um im TI-58 $n \times 10$ ($n \in \mathbb{N}_{0,6}$) Datenspeicher bereitzustellen, ist die Tastenfolge n [*Op] 17 zu betätigen.

Im Sichtfenster erscheint eine Dezimalzahl, von der die Zahl *vor* dem Punkt die letzte Programmspeicherstelle und die Zahl *nach* dem Punkt die letzte Datenspeicheradresse angibt. Z.B. erscheint nach Betätigen der Tasten

5 [*Op] 17 im Sichtfenster [79.49],

d.h. zur Verfügung stehen die 80 Programmspeicher 000 bis 079 und die 50 Datenspeicher R_{00} bis R_{49}.

Wollen wir überprüfen, welche Programm-/Datenspeicherkombination im Rechner vorhanden ist, so betätigen wir die Tastenfolge [*Op] 16. Im Sichtfenster können wir wie oben die Anzahl der Programm- und Datenspeicher ablesen.

2.2. Das Starten und Anhalten eines Programms

Wie wir ein Programm in den Rechner eingeben, haben wir soeben gelernt. Was aber haben wir zu tun, damit das Programm mit der Rechnung beginnt und an den erforderlichen Stellen zur Eingabe gegebener Werte oder zur Anzeige gesuchter Werte anhält? Hier gilt:

Mit der Anweisung [R/S] (Run Stop) wird ein laufendes Programm zum Anhalten gebracht. Der Rechner setzt seine Tätigkeit fort, wenn die Taste [R/S] manuell betätigt wird.

Sehen wir uns das Programm zur Berechnung des Flächeninhalts $A = \left(1 + \frac{\pi}{4}\right) d^2$, das wir bereits in 2.1. angegeben haben (dort ohne Erläuterungen), noch einmal an. Nebenstehend haben wir es in der üblichen Form mit Angabe der Programmspeicherstellen geschrieben. In die PSS 008 wird die Anweisung [R/S] gesetzt, um d manuell einzugeben. Danach wird das Programm manuell mit der Taste [R/S] wieder gestartet, der Rechner führt dann die Anweisung [x²] der PSS 009 aus. Nach dem Gleichheitszeichen wird erneut gestoppt, damit der errechnete Wert A im AR abgelesen werden kann. Nach dem manuellen Start mit [R/S] findet der Rechner in der PSS 012 die Anweisung [RST]. Damit wird das BAR auf die Speicherstelle 000 gestellt, der Rechenablauf kann wieder neu beginnen.

PSS	Taste
000	(
001	1
002	+
003	•π
004	÷
005	4
006	)
007	×
008	R/S
009	x²
010	=
011	R/S
012	RST

Hätten wir die letzte Anweisung [RST] *nicht* in das Programm heineingenommen, so würde der Rechner in der PSS 012 keine Anweisung vorfinden, d.h. in diesem Programmschritt wird nichts von ihm verlangt. Der Rechner geht dann zur PSS 013 über, in der wieder keine Anweisung steht, auch in der PSS 014 nicht usw., bis hin zur letzten (z.B. 239. im Normalfall beim TI-58) PSS. Hier gibt der Rechner die Suche nach weiterer Beschäftigung auf und signalisiert Unzufriedenheit. Probieren Sie selbst einmal das Programm ohne diese letzte Anweisung [RST]. Oder sollte Ihr Rechner mit seiner Untätigkeit doch zufrieden sein? Dann hat Ihr Rechner in den Speicherstellen nach 012 wohl noch Reste eines alten Programms vorgefunden und verarbeitet. Wollen Sie dieses vermeiden, so beachten Sie:

> Die manuelle Betätigung der Taste [•CP] (Clear Program) in der Betriebsart RECHNEN löscht alle Anweisungen in den Programmspeichern.

→ im Programm s.S. 40 (CP löscht T-Speicher)

Dieselbe Wirkung können Sie natürlich erreichen, indem Sie Ihren Rechner aus- und wieder einschalten.

Tasten Sie jetzt das obige Programm nach der Eingabevorschrift (s. 2.1.) in den Rechner ein. Überprüfen Sie mit diesem Programm die nebenstehende Tabelle für gegebene Durchmesser d und berechnete Flächeninhalte A. Mit [•Fix] 4 haben wir die Anzeige für A auf 4 Nachkommastellen eingestellt.

d in cm	A in cm²
2	*7,1416*
2,83	*14,2991*
4,15	*30,7490*
9,32	*155,0840*
12,80	*292,5196*
1,43	*3,6510*
0,82	*1,2005*
0,05	*0,0045*

Im folgenden geben wir noch weitere Varianten für ein Programm zur Berechnung des Flächeninhalts. Die letzten beiden Programme benutzen die Darstellung $A = d^2 \left(1 + \frac{\pi}{4}\right)$.

In allen drei Fällen kommen wir mit *einer* [R/S]-Anweisung aus. Überlegen Sie, was vom Rechner bei der Durchführung der Programme gemacht wird und was bei der ersten und den weiteren Eingaben von d vom Benutzer auszuführen ist.

PSS	Taste	Taste	Taste
000	(	R/S	x^2
001	1	x^2	x
002	+	x	(
003	•π	(	1
004	÷	1	+
005	4	+	•π
006	)	•π	÷
007	=	÷	4
008	R/S	4	)
009	x^2	)	=
010	x	=	R/S
011	RST	RST	RST

Zur Übung bringen wir noch ein weiteres einfaches

Beispiel: Berechnung von Summenwerten

$$s_m = \sum_{k=1}^{m} k^2 = 1^2 + 2^2 + 3^2 + \ldots + (m-1)^2 + m^2 \quad \text{für} \quad m \in \mathbb{N}_n .$$

Wir berechnen die Summe der ersten m Quadratzahlen nach der Formel

$s_m = s_{m-1} + m^2$ mit $m \in \mathbb{N}_n$ und $s_0 = 0$.

Wir können auch sagen:

neuer Summenwert = alter Summenwert + nächstes Glied der Reihe.

Hierfür wollen wir die aus dem ALGOL 60 (einer Programmiersprache für elektronische Rechenanlagen) bekannte Schreibweise

$$s := s + m^2$$

benutzen. (Mit dem in der Mathematik üblichen Gleichheitszeichen ist dagegen $s = s + m^2$ sinnlos.) Das Symbol := lesen wir ‚ergibt sich aus' oder ‚wird ersetzt durch' und nennen es die **Ergibtanweisung.**

Steht z.B. der alte s-Wert (s_{m-1}) im Speicher R_2 und der Wert m in R_1, so lautet für $s := s + m^2$ (wenn der neue s-Wert (s_m) zum Schluß in R_2 und im AR stehen soll) die Tastenfolge:

[RCL] 1 [x^2] [SUM] 2 [RCL] 2

oder auch: [RCL] 2 [+] [RCL] 1 [x^2] [=] [STO] 2 (umständlicher!).

Das Programm zur Berechnung der Summenwerte schreiben wir jetzt folgendermaßen:

PSS	Taste	Bemerkungen
000	1	
001	SUM	$m := m + 1 \rightarrow R_1$
002	1	
003	RCL	
004	1	$m = (R_1) \rightarrow AR$
005	x^2	m^2
006	SUM	$s := s + m^2 \rightarrow R_2$
007	2	
008	RCL	
009	2	$s = (R_2) \rightarrow AR$
010	R/S	
011	RST	Sprung auf PSS 00

Vor der Benutzung dieses Programms löschen wir mit [*CM$_s$] vorsorglich alle Datenspeicher. Die Zahlenrechnung (für n = 12) ergibt:

m	1	2	3	4	5	6	7	8	9	10	11	12
s_m	1	5	14	30	55	91	140	204	285	385	506	650

(Wenn Sie übrigens s_n berechnen wollen, ohne vorher $s_1, s_2, \ldots, s_{n-1}$ zu ermitteln, dann benutzen Sie die Summenformel $s_n = \frac{n(n+1)(2n+1)}{6}$.)

2.3. Überprüfen und Korrigieren eines Programms

Jedem, der den Umgang mit einem programmierbaren Rechner lernt (und nicht nur diesem), wird es passieren, daß ein eingegebenes Programm nicht so abläuft, wie es vom Benutzer vorgesehen war. Die häufigste Fehlerquelle wird wahrscheinlich im falschen Aufbau des Programms liegen. Möglich sind aber auch Fehler, die bei der Eingabe gemacht werden.

Um ein eingegebenes Programm zu überprüfen, schalten wir den Rechner aus der Betriebsart RECHNEN mit [RST] [LRN] auf die Speicherstelle 000 in der Betriebsart LEARN. Das gespeicherte Programm können wir Schritt für Schritt auf seine Richtigkeit überprüfen.

> In der Betriebsart LEARN wird das Befehlsadreßregister durch Betätigen der Taste [SST] (Single Step) auf die folgende, durch [BST] (Back Step) auf die vorhergehende Programmspeicherstelle eingestellt.

Für das gespeicherte Programm des ersten Beispiels in 2.2. erhalten wir auf diese Art:

[RST]	[LRN]	:	*000 53*
[SST]		:	*001 01*
[SST]		:	*002 85*
[BST]		:	*001 01* usw.

So können wir mit Hilfe der Tastenkode-Tabelle (Anhang 8.1.) feststellen, ob das eingegebene mit unserem aufgestellten Programm übereinstimmt. Eventuelle Fehler können dann korrigiert werden, wie wir weiter unten sehen werden.

Die Taste [BST] wird sehr häufig auch unmittelbar bei der Eingabe eines Programms benutzt, um festzustellen, welche Taste als letzte gedrückt wurde. Bei unserem Beispiel aus 2.2. würde bei der Eingabe nach dem Programmschritt 006 (Betätigen von [)]) im Sichtfenster [007 00] erscheinen. Hieraus ist nicht mehr zu erkennen, welche Anweisung als letzte eingegeben wurde. Um zu kontrollieren, ob auch wirklich [)] eingetastet wurde, stellen wir das Befehlsadreßregister mit [BST] zurück und erhalten (bei richtiger Eingabe) [006 54].

Wenn wir die Vermutung haben, daß bei einem gespeicherten Programm z.B. im Programmschritt 047 eine Anweisung nicht korrekt ist, so wäre es natürlich sehr mühsam, von 000 mit [SST] bis 047 zu kommen. Hier hilft uns die folgende Eigenschaft der Taste [GTO]:

> Aus der Betriebsart RECHNEN wird durch die Tastenfolge [GTO] n m p [LRN] das Befehlsadreßregister auf die PSS n m p in der Betriebsart LEARN eingestellt.

(Aber aufpassen: In der Betriebsart LEARN zerstören Sie, jedenfalls teilweise, mit der Tastenfolge [GTO] n m p [LRN] Ihr Programm.)

Beseitigung von Eingabefehlern:

1. Nehmen wir an, in Ihrem (bereits gespeicherten) Programm sollte im Programmschritt 047 die Taste [÷] gedrückt worden sein, Sie vermuten aber aufgrund der ausgegebenen Ergebnisse dort einen Fehler. Sie betätigen

[GTO] 047 [LRN] und lesen die Anzeige [047 65].

Sie haben also bei der Eingabe aus Versehen die Taste [x] (Kode 65) statt [÷] (Kode 55) betätigt. In diesem Fall brauchen Sie nur bei obiger Befehlsadreßregistereinstellung die richtige Taste [÷] zu drücken. Die alte (hier: falsche) Eingabe wird dann durch die neue (hier: richtige) Eingabe überschrieben. Von der auf die PSS 048 vorgerückten Anzeige schalten Sie zurück mit [BST] und kontrollieren die Anzeige [047 55] auf Ihre Richtigkeit.

2. Bei einem Programm mögen die richtigen Programmschritte 046 bis 050 (s. Programm Dreiecksberechnung in 2.6.) wie unten links aussehen. Kontrolle des Programms ergibt dazu im Sichtfenster die Folge unten rechts.

046	4
047	÷
048	2
049	x
050	RCL

046 04
047 55
048 02
049 02
050 65

Hier ist die 2 aus Versehen zweimal eingetastet worden. In diesem Fall stellen wir das Befehlsadreßregister auf 049 ein, in der Anzeige erscheint [049 02]. Diese Operation soll gelöscht werden.

> Mit [*Del] (Delete) wird in der Betriebsart LEARN die im Sichtfenster angezeigte Anweisung gelöscht. Alle nachfolgenden Anweisungen rücken im Befehlsadreßregister um eine Stelle vor.

Stellen wir im obigen Beispiel des BAR auf 049 und betätigen [*Del], so wird die 2 in der PSS 049 gelöscht. In diese Speicherstelle rückt die nächste Anweisung [x] (Kode 65) vor. In 050 wird die Anweisung [RCL] (Kode 43) gespeichert usw.

Der Rechner bietet uns eine weitere Möglichkeit, eine Anweisung zu löschen:

> Die Anweisung [*Nop] (No Operation oder Nulloperation) veranlaßt den Rechner, diese PSS zu überspringen und die Anweisung der folgenden PSS auszuführen.

Mit [*Nop] bekommt der Rechner also die Anweisung, in dieser PSS nichts zu tun. Betätigen wir im obigen Beispiel bei der Anzeige [049 02] die Taste [*Nop], so wird der Rechner bei der Durchführung des Programms in der Betriebsart RUN nach der Anweisung in der PSS 048 die Anweisung [×] der PSS 050 ausführen.

Das Löschen einer Anweisung mit [*Nop] werden wir immer dann dem mit [*Del] vorziehen, wenn die Numerierung der folgenden Anweisungen im BAR erhalten bleiben soll, weil z.B. eine Sprunganweisung [GTO] n m p auftritt (s. 2.6.).

3. Der letzte Eingabefehler, der vorkommen kann, besteht im Vergessen des Betätigens einer Taste. Ist im obigen Beispiel im Programmschritt 047 vergessen worden, die Taste [÷] zu drücken, so würden wir mit [GTO] 046 [LRN] und Betätigen der Taste [SST] im Sichtfenster die nebenstehende Folge erhalten. In diesem Fall müssen wir in der PSS 047 Platz schaffen für die Anweisung [÷], ohne dadurch die in 047 gespeicherte 2 zu zerstören.

046 04
047 02
048 65

> Mit [*Ins] (Insert) werden in der Betriebsart LEARN alle Anweisungen von der eingestellten Befehlsadreßregisterstelle ab um eine Stelle zurückgerückt. In die frei werdende PSS kann die fehlende Anweisung gesetzt werden.

In dem obigen Beispiel werden wir bei der Einstellung des BAR auf 047 die Tasten [*Ins] [÷] betätigen und damit die richtige Anweisungsfolge erhalten (s. nebenstehend).

046 04
047 55
048 02
049 65

Eine weitere Bemerkung zur Taste [SST]

Haben wir dem Rechner ein Programm eingegeben und wieder in die Betriebsart RECHNEN geschaltet, so kann die Ausführung dieses Programms schrittweise verfolgt werden.

> Mit der Taste [SST] wird in der Betriebsart RECHNEN jede Anweisung des Programms einzeln durchgeführt.

Geben Sie das kurze Programm aus 2.2. in Ihren Rechner. Sehen Sie sich das Sichtfenster an, wenn Sie wiederholt die Taste [SST] betätigen. Sie erhalten die nebenstehende Anzeigenfolge. An der PSS 008 [R/S] geben Sie 2 ein.

Diese Eigenschaft der Taste [SST] wird vorwiegend zum Aufsuchen von Fehlerquellen in einem Programm benutzt.

0.
1
1.
3.141592654
3.141592654
4
1.785398163
1.785398163
2
2
4.
7.141592654

2.4. Programmadreßtasten

In den ersten Programmbeispielen mußten wir vor dem Starten des Programms das Befehlsadreßregister mit [RST] auf die PSS 000 stellen. Bei vielen Aufgaben soll das Programm von einer anderen PSS ab durchgeführt werden. Dieses können wir mit einer der 10 Programmadreßtasten [A] bis [E] und [•A'] bis [•E'] erreichen. Als Platzhalter für eine dieser Tasten schreiben wir [M] (Marke).

> Mit einer Programmadreßtaste [M] wird das Programm in der Betriebsart RUN von der PSS ab gestartet, die auf [•LBL] [M] (Label [M]) im Programm folgt. [M] kann sowohl manuell als auch als Anweisung in einem Programm benutzt werden.

Mit den Programmadreßtasten können wir somit 10 bestimmte Befehlsadreßregisterstellen im Programm markieren.

Beispiel: Für die in den ersten zwei Zeilen der Tabelle 2.4.1 (weiter unten) gegebenen nicht-negativen Werte a und b sollen

$$u = \left(\frac{a}{b+1} + b\right)\sqrt{9+a^2} \quad \text{und} \quad v = \frac{2+\sqrt{a}+\sqrt{b}}{1+u^2}$$

berechnet werden.

Speicherplan	
1	a
2	b
3	u
4	v

In einem Speicherplan (Registerplan) legen wir zunächst fest, in welche Speicher wir die gegebenen und berechneten Werte bringen werden (s. nebenstehende Tabelle). Die Werte a und b sollen mit den Programmadreßtasten [A] und [B] in die Speicher gebracht werden. Es bedeutet z.B. a → [A]: Der gegebene Wert a wird eingetastet (ins AR gebracht), und nach Betätigen der Taste [A] werden die im Programm auf [•LBL] [A] folgenden Anweisungen ausgeführt (hier: a wird nach R_1 gespeichert). Entsprechend sollen die berechneten Werte u und v mit den Programmadreßtasten [D] und [E] abgerufen werden.

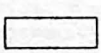
[D] → u: Durch Betätigen der Taste [D] werden die auf [•LBL] [D] folgenden Anweisungen ausgeführt und zum Schluß der Wert u ins AR gebracht.

Zunächst stellen wir uns den Eingabe-, Rechen-, Speicher- und Ausgabeablauf schematisch durch ein **Flußdiagramm** oder einen **Programmablaufplan** dar. Dabei verwenden wir die folgenden Symbole:

für Labels, Marken, Kennzeichen;

für Eingabe oder Ausgabe;

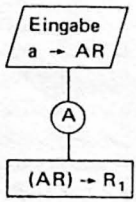
für Start oder Stop;



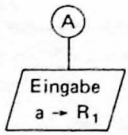
für Berechnungen oder Speicheranweisungen.

Das *Flußdiagramm* für unsere Aufgabe kann dann folgendermaßen aussehen, wobei wir

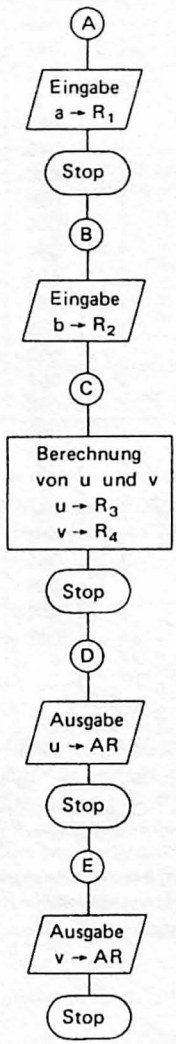
statt



kurz



zeichnen.



Aus dem Flußdiagramm entwickeln wir das Programm:

PSS	Taste	Bemerkungen
000	*LBL	
001	A	
002	STO	a → R_1
003	1	
004	R/S	
005	*LBL	
006	B	
007	STO	b → R_2
008	2	
009	R/S	
010	*LBL	Beginn der
011	C	Rechnung
012	RCL	
013	1	
014	÷	
015	(	
016	RCL	b → AR
017	2	
018	+	
019	1	
020	)	b + 1
021	+	a/(b + 1)
022	RCL	b → AR
023	2	
024	=	a/(b + 1) + b
025	×	
026	(	
027	9	
028	+	
029	RCL	a → AR
030	1	
031	x^2	a^2
032	)	$9 + a^2$
033	$\sqrt{x}$	$\sqrt{9 + a^2}$
034	=	u
035	STO	u → R_3
036	3	
037	x^2	u^2
038	+	
039	1	
040	=	$1 + u^2$
041	1/x	$1/(1 + u^2)$
042	×	
043	(	
044	2	
045	+	
046	RCL	a → AR
047	1	
048	$\sqrt{x}$	$\sqrt{a}$
049	+	$2 + \sqrt{a}$
050	RCL	b → AR
051	2	
052	$\sqrt{x}$	$\sqrt{b}$
053	)	$2 + \sqrt{a} + \sqrt{b}$
054	=	v
055	STO	v → R_4
056	4	
057	R/S	
058	*LBL	
059	D	
060	RCL	u → AR
061	3	
062	R/S	(AR) = u
063	*LBL	
064	E	
065	RCL	v → AR
066	4	
067	R/S	(AR) = v

Diesem Programm fügen wir eine Anleitung hinzu, aus der hervorgeht, was der Benutzer *nach* der Eingabe des Programms in den Rechner zu beachten hat. Diejenigen Leser, die sich selbst eine Programmbibliothek aufbauen wollen, sollten jedes Programm nur mit einer dazugehörenden **Benutzeranleitung** aufbewahren. Erfahrungsgemäß vergißt man nach einiger Zeit die Einzelheiten des Programms und weiß z.B. nicht mehr genau, in welcher Reihenfolge gegebene Werte eingetastet oder gesuchte Werte angezeigt werden. Für viele Aufgaben ist auch die Kenntnis des Speicherplans wichtig, deshalb nehmen wir diesen ebenfalls mit in unsere Anleitungsvorschrift hinein.

Folgende Punkte werden wir stets bei der Zusammenstellung einer Benutzeranleitung beachten:

1. Das Programm wird nach der Vorschrift 2.1. eingegeben, wobei 4. immer dann fehlen darf, wenn das Programm mit einer Programmadreßtaste gestartet wird.
2. Striche in der Eingabe- oder Anzeigespalte bedeuten, daß keine Eingabe erfolgt oder der angezeigte Wert im AR ohne Bedeutung ist.
3. Im Speicherplan nicht aufgeführte Speicher sind nicht belegt, stehen also für evtl. Programmerweiterungen zur Verfügung.

Für unser obiges Beispiel geben wir die *Benutzeranleitung:*

Berechnung von

$$u = \left(\frac{a}{b+1} + b\right)\sqrt{9+a^2} \quad \text{und} \quad v = \frac{2+\sqrt{a}+\sqrt{b}}{1+u^2}$$

für gegebene Werte a und b.

Speicherplan		Eingabe	Taste	Anzeige
1	a	a	A	–
2	b	b	B	–
3	u	–	C	–
4	v	–	D	u
		–	E	v

Wir haben dieses Programm sehr aufwendig geschrieben. Die Reihenfolge bei der Eingabe und der Ausgabe ist hier beliebig, d.h. wir können b vor a eingeben und auch v vor u ausgeben. Das wird bei späteren Programmen, die ohne Programmadreßtasten geschrieben werden, nicht der Fall sein.

Das Programm testen wir mit den Werten $a = b = 0$ (hierfür erhalten wir $u = 0$ und $v = 2$) und $a = 4$ und $b = 1$ ($u = 15$ und $v = \frac{5}{226} = 0{,}02212$). Nach erfolgreichem Test berechnen wir die Werte u und v in der 3. und 4. Zeile der Tabelle 2.4.1 auf 5 Nachkommastellen.

Tabelle 2.4.1

a	2	1,2	4,684	0,84314	12,60023
b	3	2,8	3,125	1,75020	8,74105
u	12,61943	10,06742	23,69852	6,40938	129,97219
v	0,03211	0,04659	0,01054	0,10079	0,00050

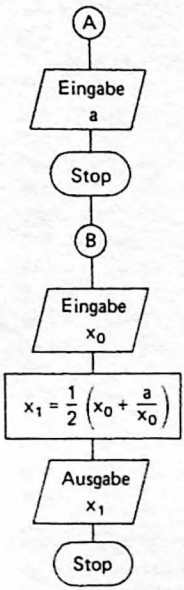
Im II. Teil dieses Buches werden wir Programmbeispiele aus verschiedenen Gebieten der Mathematik und Technik bringen. Damit aber das unmittelbar Gelernte der Programmiertechnik geübt und gefestigt wird, sollen bereits hier einige weitere Beispiele gebracht werden.

Beispiel: Die Quadratwurzel $x = \sqrt{a}$ ist nach der Iterationsvorschrift (s. 1.3.)

$$x_n = \frac{1}{2}\left(x_{n-1} + \frac{a}{x_{n-1}}\right) \quad \text{für} \quad n \in \mathbb{N}$$

zu berechnen.

Im *Flußdiagramm* setzen wir $x_{n-1} = x_0$ und $x_n = x_1$ (unten links). Danach stellen wir mit $a \rightarrow R_1$ und $x_0 \rightarrow R_2$ das Programm (unten rechts) auf. (Auf die früher angefügten Bemerkungen verzichten wir.)



PSS	Taste
000	*LBL
001	A
002	STO
003	1
004	R/S
005	*LBL
006	B
007	STO
008	2
009	+
010	RCL
011	1
012	÷
013	RCL
014	2
015	=
016	÷
017	2
018	=
019	R/S

Benutzeranleitung:

Quadratwurzelberechnung $x = \sqrt{a}$ durch Iteration				
Speicherplan		Eingabe	Taste	Anzeige
1	a	a	A	–
2	x_0	x_0	B	x_1
		–	B	x_2
		...	usw.	...

Führen Sie die Rechnung durch für $a = 14{,}256$ mit $x_0 = 3$. Sie erhalten die Folge:

3,876; 3,777009288; 3,775712075; 3,775711853; 3,775711853; ...

Wählen Sie einen anderen Ausgangswert x_0, und führen Sie mit diesem die Rechnung durch. Sie kommen stets auf 3,775711853, d.h. die durch die Iterationsvorschrift festgelegte Folge ist für jeden Wert x_0 konvergent. Natürlich kann die Anzahl der Iterationsschritte – und damit der Zeitaufwand – sehr groß werden. Wählen Sie z.B.

$x_0 = 500\,000\,000$ oder $x_0 = 0{,}0001$.

Beispiel: Sind von einer quadratischen Funktion

$y = a x^2 + b x + c$

drei Funktionswerte y_1, y_2, y_3 an den äquidistanten Stellen x_1, $x_2 = x_1 + h$, $x_3 = x_2 + h$ (s. Bild 2.4.1) bekannt, so kann der Funktionswert y_4 an der Stelle $x_4 = x_3 + h$ nach der Formel

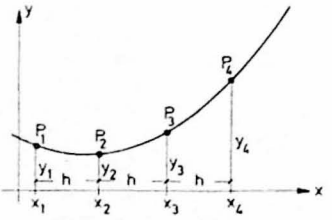


Bild 2.4.1

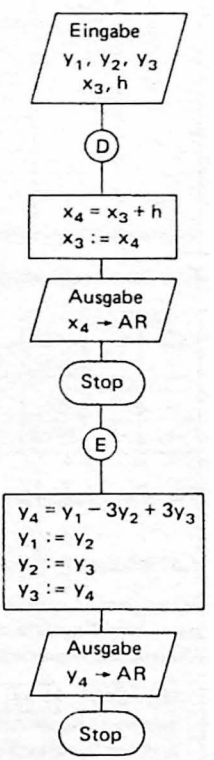
$$y_4 = y_1 - 3y_2 + 3y_3$$

berechnet werden.[1)]

Sind z.B. y_1, y_2, y_3 gemessen worden, so lassen sich die weiteren Funktionswerte $y_4, y_5, \ldots$ an äquidistanten Stellen $x_4, x_5, \ldots$ berechnen, ohne daß erst die Funktionsgleichung aufgestellt zu werden braucht.

Zur Berechnung der zugehörigen Werte x und y entwickeln wir das *Flußdiagramm*, wobei wir die Eingabe nur grob skizzieren. Die Zuordnung zwischen den Daten y, x, h und den Speichern oder Programmadreßtasten ist in der Benutzeranleitung festgelegt. Wir erinnern vorher auch noch einmal an die Ergibtanweisung (s. 2.2.). Es bedeutet

$y_1 := y_2$: Der Wert y_1 wird ersetzt durch y_2,
oder: y_2 wird in den Speicher gebracht,
in dem sich bisher y_1 befand.



[1)] Für mathematisch Interessierte: Versuchen Sie, diese einfache Formel zur Berechnung von y_4 aus den vorhergehenden Funktionswerten y_1, y_2, y_3 allgemein zu beweisen.

Das Programm lautet (Speicherplan s. Benutzeranleitung):

PSS	Taste
000	*LBL
001	A
002	STO
003	1
004	R/S
005	*LBL
006	B
007	STO
008	2
009	R/S
010	*LBL
011	C
012	STO
013	3

014	R/S
015	*LBL
016	*A'
017	STO
018	4
019	R/S
020	*LBL
021	*B'
022	STO
023	5
024	R/S
025	*LBL
026	D
027	RCL
028	5

029	SUM
030	4
031	RCL
032	4
033	R/S
034	*LBL
035	E
036	RCL
037	1
038	−
039	3
040	×
041	RCL
042	2
043	STO

044	1
045	+
046	3
047	×
048	RCL
049	3
050	STO
051	2
052	=
053	STO
054	3
055	R/S

Benutzeranleitung:

Funktionswerte einer Parabel $y = ax^2 + bx + c$

Speicherplan		Eingabe	Taste	Anzeige
1	y_1	y_1	A	–
2	y_2	y_2	B	–
3	y_3	y_3	C	–
4	x_3	x_3	*A'	–
5	h	h	*B'	–
		–	D	x_4
		–	E	y_4
		–	D	x_5
		–	E	y_5
		...	usw.	...

Zwei Zahlenbeispiele (die Werte der ersten 3 Spalten sind gegeben):

x	−1	0	1	2	3	4	5	6	7	8	9	10
y	0,5	0	0,5	2	4,5	8	12,5	18	24,5	32	40,5	50

x	0	0,25	0,5	0,75	1	1,25	1,5	1,75	2
y	2	1,8125	1,75	1,8125	2	2,3125	2,75	3,3125	4

(Die quadratischen Funktionen für die obigen Beispiele lauten: $y = \frac{x^2}{2}$ oder $y = x^2 - x + 2$.)

2.5. Indirekte Speicheradressierung

Bei der letzten Aufgabe in 2.4. nahm die Eingabe von y_1, y_2, y_3, x_3 und h 25 Programmspeicherplätze in Anspruch. Diese Zahl können wir mit einer indirekten Speicheradressierung wesentlich reduzieren.

Mit [STO] [*Ind] n m wird der Inhalt des Anzeigeregisters in den Speicher gebracht, dessen Adresse in R_{nm} steht. Eine entsprechende Regel gilt für die anderen Speichertasten [RCL], [SUM], [*Exc], [*Prd].

Ist z.B. $(R_{15}) = 21$, so wird durch die Tastenfolge

314.56 [STO] [*Ind] 15

der Zahlenwert (AR) = 314.56 nach R_{21} gespeichert.

Wichtig: Für [STO] [*Ind] wird nur eine Programmspeicherstelle benötigt. Auch bei der indirekten Speicheradressierung darf die Kurzform (s. 1.3.) gewählt werden.

PSS	Taste
000	*LBL
001	A
002	STO *Ind
003	00
004	1
005	SUM
006	0
007	R/S

Das Programm zur Eingabe der Werte y_1 bis h des letzten Beispiels in 2.4. ist mit der indirekten Speicherung nebenstehend aufgezeichnet. Statt 25 benötigen wir nur noch 8 Programmspeicherplätze. Die Eingabevorschrift lautet:

[*CM$_s$] 0 [A] y_1 [A] y_2 [A] y_3 [A] x_3 [A] h [A].

Die 0 vor der 1. Programmadreßtaste [A] ist erforderlich, damit bei der indirekten Speicherung von y_1 nach R_1 die richtige Adresse – hier 1 – im Speicher R_0 vorgefunden wird. Danach wird vor der Eingabe des nächsten Wertes mit 1 [SUM] 0 der Inhalt von R_0 um jeweils 1 erhöht. (Bei SUM darf die Kurzformadressierung gewählt werden, in der PSS 003 wegen der folgenden 1 jedoch nicht.)

Beispiel: Wir wollen ein Programm zur Lösung des linearen Gleichungssystems

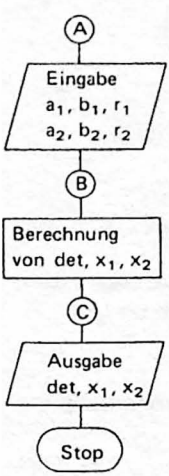
$a_1 x_1 + b_1 x_2 = r_1$

$a_2 x_1 + b_2 x_2 = r_2$

schreiben. Es gilt (für $\det \neq 0$, was hier vorausgesetzt werden soll):

$$\det = a_1 b_2 - a_2 b_1, \quad x_1 = \frac{r_1 b_2 - r_2 b_1}{\det}, \quad x_2 = \frac{a_1 r_2 - a_2 r_1}{\det}.$$

Die Zahlenwerte det, x_1 und x_2 sollen vom Rechner angezeigt werden. Wir skizzieren das *Flußdiagramm* und legen den Speicherplan fest.



Speicherplan	
0	1, ..., 10
1	a_1
2	b_1
3	r_1
4	a_2
5	b_2
6	r_2
7	–
8	det
9	x_1
10	x_2

Die Beendigung der Rechnung soll durch eine 0 im AR angezeigt werden.

PSS	Taste
000	*LBL
001	A
002	STO *Ind
003	00
004	1
005	SUM
006	0
007	R/S
008	*LBL
009	B
010	RCL
011	1
012	x
013	RCL
014	5
015	−
016	RCL

017	2
018	x
019	RCL
020	4
021	=
022	STO
023	8
024	RCL
025	3
026	x
027	RCL
028	5
029	−
030	RCL
031	6
032	x
033	RCL
034	2

035	=
036	÷
037	RCL
038	8
039	=
040	STO
041	9
042	RCL
043	1
044	x
045	RCL
046	6
047	−
048	RCL
049	4
050	x
051	RCL
052	3

053	=
054	÷
055	RCL
056	8
057	=
058	STO
059	10
060	0
061	R/S
062	*LBL
063	C
064	1
065	SUM
066	0
067	RCL *Ind
068	0
069	R/S
070	

Wir schreiben die *Benutzeranleitung* in folgender Form:

Eingabe: [*CM$_s$] 0 [A] a_1 [A] b_1 [A] r_1 [A] a_2 [A] b_2 [A] r_2 [A];

Berechnung: [B];

Ausgabe: [C] det [C] x_1 [C] x_2.

Wir testen das Programm mit den Gleichungssystemen

$$\begin{aligned} 2x_1 + 3x_2 &= 8 \\ x_1 + 4x_2 &= 9 \end{aligned}$$ mit $\det = 5$ und der Lösung $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ und

$$\begin{aligned} -3x_1 + 7x_2 &= 1 \\ x_1 - 6x_2 &= -4 \end{aligned}$$ mit $\det = 11$ und der Lösung $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$.

Nach diesem Test erhalten wir z.B. für (mit [*Fix] 3)

$$\begin{aligned} 3{,}74\,x_1 + 5{,}61\,x_2 &= 16{,}72 \\ 1{,}89\,x_1 + 4{,}53\,x_2 &= 12{,}34 \end{aligned}$$: $\det = 6{,}339$ und $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1{,}028 \\ 2{,}295 \end{pmatrix}$.

Bemerkungen zur indirekten Speicheradressierung [STO] [*Ind] n m:

1. Stehen im Rechner s Datenspeicher zur Verfügung (z.B. s = 30 beim Einschalten des TI-58) und ist $(R_{nm}) \geq s$, so wird dieses durch Blinken angezeigt.
2. Steht in R_{nm} eine positive Dezimalzahl, so gibt der ganzzahlige Anteil dieser Zahl – sofern er kleiner als s ist, sonst siehe 1. – die Speicheradresse an, in die (AR) gebracht wird.
3. Ist $(R_{nm}) \leq 0$, so wird (AR) nach R_0 gespeichert.

Ist z.B. $(R_{12}) = 25{,}489$, so wird mit [STO] [*Ind] 12 der Inhalt des Anzeigeregisters nach R_{25} gebracht.

Für die übrigen Speichertasten gelten die entsprechenden Bemerkungen.

2.6. Die unbedingte Sprunganweisung GTO

Von einem Dreieck sind zwei Winkel
$\alpha = 47{,}5°$ und $\beta = 62{,}8°$ gegeben.
Für die Seite

a = 13,5; 18,2; 21,4; 27,8; 32,0 cm

sind die Seiten b und c, der Flächeninhalt A und der Umfang U zu berechnen.

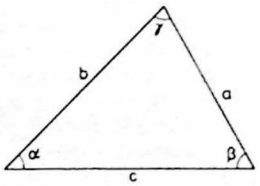


Bild 2.6.1

Es gilt:

$\gamma = 180° - (\alpha + \beta)$ (Winkelsumme im Dreieck = 180°),

$\frac{a}{\sin\alpha} = \frac{b}{\sin\beta} = \frac{c}{\sin\gamma}$ (Sinussatz der ebenen Trigonometrie),

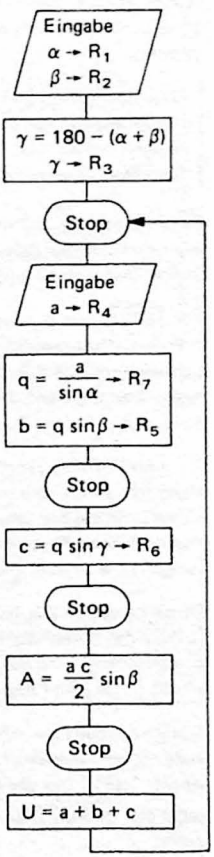
$A = \frac{1}{2} c\, h_c = \frac{a\,c}{2} \sin\beta$ und $U = a + b + c$.

Den Sinussatz schreiben wir:

$b = q \sin\beta$ und $c = q \sin\gamma$ mit $q = \frac{a}{\sin\alpha}$.

Wir wollen das Programm ohne Benutzung der Programmadreßtasten schreiben.

Das *Flußdiagramm* kann folgendermaßen aussehen:



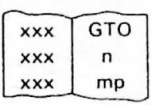
Die Stop-Anweisungen unterbrechen den Programmablauf, um gegebene Werte (α, β, a) einzugeben oder gesuchte Werte (γ, b, c, A, U) anzuzeigen. Die Fortsetzung des Programmablaufs erreichen wir manuell mit der Taste [R/S]. Nach der Berechnung von U haben wir zwei Ziele vor Augen: 1. Wir wollen den Wert U im AR ablesen, und 2. soll danach das Programm mit der Eingabe des nächsten a-Wertes wieder gestartet werden. Im Flußdiagramm haben wir dieses schematisch dargestellt durch eine Rückkehr auf (Stop) vor der Eingabe von a. Diese Stelle kann im Programm entweder durch ihre dreiziffrige PSS n m p oder durch [•LBL] [M], wo [M] wie früher als Platzhalter für eine Marke steht, angegeben werden.

> Als Marke (Label, Kennzeichen) [M] dürfen in einem Programm alle Tasten bis auf [2nd], [LRN], [•Ins], [•Del], [SST], [BST], [•Ind], [R/S] und die Zifferntasten [1] bis [9] benutzt werden.

In 2.4. haben wir die Programmadreßtasten bereits als Marken kennengelernt. Mit diesen Sondermarken [A] bis [•E'] kann ein Programm manuell von der durch dieses Label markierten Stelle ab gestartet werden. Dieses ist mit den übrigen Marken nicht möglich. Wir dürfen z.B. nicht in dem 1. Beispiel in 2.4. [•LBL] [A] durch [•LBL] [x^2] ersetzen. Diese Marken werden hauptsächlich durch Anweisungen in einem Programm aufgerufen.

> Die Anweisung [GTO] n m p oder [GTO] [M] bewirkt in einem Programm einen Sprung auf die PSS n m p oder auf die im Programm auf [•LBL] [M] folgende PSS und die Ausführung der in dieser Programmspeicherstelle festgelegten Anweisung. Ist [M] eine Programmadreßtaste, so darf [GTO] fehlen.

Eine Anweisung, mit der wir eine beliebige Stelle im Programm erreichen können, nennen wir eine Sprunganweisung. [GTO] (Go To) ist eine **unbedingte Sprunganweisung**, da sie immer (unbedingt) ausgeführt werden soll.



Für [GTO] n m p werden nach nebenstehender Aufzeichnung drei Programmspeicherplätze benötigt. Bei der **Kurzformadressierung** darf z.B. für [GTO] oop auch [GTO] p eingetastet werden, wenn die auf p folgende Taste **keine** Zifferntaste ist (s. 1.3.). Für die Speicherung werden auch hier drei Programmspeicherplätze benötigt.

Ob man in einem Programm lieber mit [GTO] n m p oder [GTO] [M] arbeitet, hängt sicherlich etwas vom persönlichen Geschmack des Programmierers ab. Einfacher und sicherer ist die Benutzung eines Labels, denn im anderen Fall ist die PSS n m p beim Aufstellen des Programms oft noch nicht bekannt oder ändert sich bei Programmänderungen (z.B. durch [•Del] oder [•Ins], s. 2.3.) ebenfalls.

Bemerkung: In 2.3. haben wir aus der Betriebsart RECHNEN manuell mit [GTO] n m p [LRN] das Befehlsadreßregister auf die PSS n m p in der Betriebsart LEARN eingestellt. Entsprechend wird mit [GTO] [M] [LRN] das BAR auf die PSS eingestellt, die auf [•LBL] [M] im Programm folgt.

Übrigens haben wir mit [RST] früher (s. 2.1.) bereits eine andere unbedingte Sprunganweisung kennengelernt. Während wir aber mit [GTO] jede Speicherstelle erreichen können, ist mit [RST] nur der Sprung auf die PSS 000 (also auf den Anfang des Programms) möglich. [GTO] 000 bewirkt dasselbe wie [RST], benötigt aber 2 Programmspeicher mehr.

Aus dem Flußdiagramm entwickeln wir das Programm für unsere Dreiecksberechnungen:

PSS	Taste	Bemerkungen
000	STO	$\alpha \rightarrow R_1$
001	1	
002	R/S	
003	STO	$\beta \rightarrow R_2$
004	2	
005	+	
006	RCL	$\alpha \rightarrow$ AR
007	1	Berechnung
008	−	von γ
009	1	
010	8	
011	0	
012	=	
013	+/−	
014	STO	$\gamma \rightarrow R_3$
015	3	
016	R/S	(AR) = γ, (AR) = U
017	STO	$a \rightarrow R_4$
018	4	
019	÷	Berechnung
020	RCL	von q
021	1	$\alpha \rightarrow$ AR
022	*sin	
023	=	
024	STO	$q \rightarrow R_7$
025	7	Berechnung
026	×	von b
027	RCL	$\beta \rightarrow$ AR
028	2	
029	*sin	
030	=	
031	STO	$b \rightarrow R_5$
032	5	
033	R/S	(AR) = b
034	RCL	$\gamma \rightarrow$ AR
035	3	
036	*sin	Berechnung
037	×	von c
038	RCL	$q \rightarrow$ AR
039	7	
040	=	
041	STO	$c \rightarrow R_6$
042	6	
043	R/S	(AR) = c
044	×	
045	RCL	$a \rightarrow$ AR
046	4	
047	÷	Berechnung
048	2	von A
049	×	
050	RCL	$\beta \rightarrow$ AR
051	2	
052	*sin	
053	=	
054	R/S	(AR) = A
055	RCL	$a \rightarrow$ AR
056	4	
057	+	
058	RCL	$b \rightarrow$ AR
059	5	Berechnung
060	+	von U
061	RCL	$c \rightarrow$ AR
062	6	
063	=	
064	GTO	Sprung auf
065	0	PSS 016
066	16	

Benutzeranleitung:

Berechnung eines Dreiecks aus α, β, a

1. Programm eintasten.
2. Vor der Durchführung der Rechnung mit neuen Werten α und β ist das BAR mit [RST] auf 000 zu stellen.

Speicherplan		Eingabe	Taste	Anzeige
1	α	α	R/S	−
2	β	β	R/S	γ
3	γ	a	R/S	b
4	a	−	R/S	c
5	b	−	R/S	A
6	c	−	R/S	U
7	q	a	R/S	b
		...	usw.	...

Die Ergebnisse der Rechnung sind in der folgenden Tabelle zusammengestellt (mit [*Fix] 1)

Dreiecksberechnungen: $\alpha = 47{,}5°$, $\beta = 62{,}8°$, $\gamma = (R_3) = 69{,}7°$					
a in cm	13,5	18,2	21,4	27,8	32,0
b in cm	16,3	22,0	25,8	33,5	38,6
c in cm	17,2	23,2	27,2	35,4	40,7
A in cm^2	103,1	187,4	259,1	437,2	579,3
U in cm	47,0	63,3	74,4	96,7	111,3

Beispiel: Berechnung von Flächenwerten.

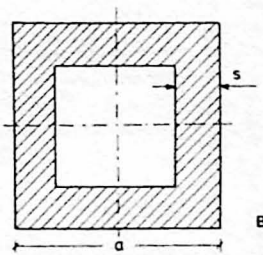


Bild 2.6.2

Für einen Träger mit quadratischen Rohrquerschnitt (s. Bild 2.6.2) sind für Festigkeitsberechnungen die folgenden Größen wichtig:

Flächeninhalt

$$A = a^2 - (a - 2s)^2 = 4s\,(a - s);$$

Flächenträgheitsmoment

$$I = \frac{1}{12}\,[a^4 - (a - 2s)^4] = \frac{A}{12}\,[a^2 + (a - 2s)^2];$$

Widerstandsmoment

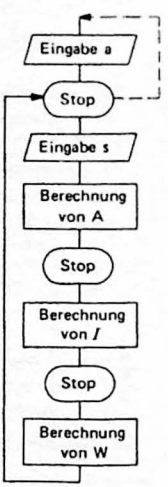
$$W = \frac{I}{a/2} = \frac{2I}{a}.$$

Wir wollen die Berechnung dieser Größen für handelsübliche Abmessungen durchführen:

a = 35 mm;	s = 1,25;	1,5;	1,75;	2;	2,25;	2,5	mm
a = 40 mm;	s = 1,5;	2;	2,5				mm
a = 50 mm;	s = 2;	2,5;	3;	3,5;	4		mm

Die Ergebnisse A in cm^2, I in cm^4 und W in cm^3 sollen auf 2 Nachkommastellen ausgegeben werden.

Im *Flußdiagramm* wird nach der Durchführung der Rechnung für ein Zahlenpaar a, s auf die Eingabestelle für s zurückgegangen. Soll auch ein neuer Wert a eingegeben werden, so stellen wir manuell mit [RST] (im Flußdiagramm gestrichelte Linie) das Befehlsadreßregister auf 000.



Das Programm lautet:

PSS	Taste
000	STO
001	1
002	R/S
003	STO
004	2
005	×
006	4
007	×
008	(
009	RCL
010	1

011	−
012	RCL
013	2
014	)
015	=
016	R/S
017	÷
018	1
019	2
020	×
021	(
022	RCL

023	1
024	x^2
025	+
026	(
027	RCL
028	1
029	−
030	2
031	×
032	RCL
033	2
034	)

035	x^2
036	=
037	R/S
038	×
039	2
040	÷
041	RCL
042	1
043	=
044	GTO
045	0
046	02

Benutzeranleitung:

Berechnung von Flächenwerten

1. Programm eintasten.
2. Vor der Eingabe eines neuen Wertes a mit RST das BAR auf 000 stellen.

Speicherplan		Eingabe	Taste	Anzeige
1	a	a	R/S	−
2	s	s	R/S	A
		−	R/S	*I*
		−	R/S	W
		s	R/S	A
		...	usw.	...

Die Ergebnisse der Rechnung sind in der folgenden Tabelle zusammengestellt:

a in mm	s in mm	A in cm^2	*I* in cm^4	W in cm^3
35	1,25	1,69	3,21	1,83
	1,5	2,01	3,77	2,15
	1,75	2,33	4,30	2,46
	2	2,64	4,81	2,75
	2,25	2,95	5,29	3,03
	2,5	3,25	5,76	3,29
40	1,5	2,31	5,72	2,86
	2	3,04	7,34	3,67
	2,5	3,75	8,83	4,41
50	2	3,84	14,77	5,91
	2,5	4,75	17,91	7,16
	3	5,64	20,85	8,34
	3,5	6,51	23,59	9,44
	4	7,36	26,15	10,46

2.7. Übungsaufgaben

2.1. Schreiben Sie ein Programm zur Berechnung der Flächenwerte eines kreisförmigen Querschnitts:

Flächeninhalt $A = \frac{\pi}{4} d^2$;

Widerstandsmoment $W = \frac{\pi}{32} d^3$;

Flächenträgheitsmoment $I = \frac{\pi}{64} d^4$.

(Beachten Sie, daß W und I über d mit A zusammenhängen. Sie sparen sich dadurch einige Programmschritte!)
Führen Sie die Rechnung durch für

d = 20; 25; 30; 35; 40; 45; 50 mm .

2.2. Von einem Dreieck sind gegeben: a = 56,4 cm; b = 38,2 cm;

γ = 17,8°; 48,9°; 81,0°; 104,7°; 126,5°; 146,1° .

Berechnen Sie mit einem Programm die Seite c nach dem Cosinussatz:

$c^2 = a^2 + b^2 - 2ab \cos\gamma$.

2.3. Die 3. Wurzel aus einer Zahl a läßt sich iterativ nach der Vorschrift

$$x_n = \frac{1}{3}\left(2x_{n-1} + \frac{a}{x_{n-1}^2}\right) \quad \text{mit} \quad n \in \mathbb{N}$$

berechnen. Stellen Sie hierfür ein Programm auf, und berechnen Sie insbesondere $\sqrt[3]{6{,}42}$ und $\sqrt[3]{-0{,}0386}$. Vergleichen Sie Ihre iterativ berechneten Werte mit den über die Tasten [INV] [y^x] angezeigten Werten.

Zusatz: Untersuchen Sie die Folge

$$x_n = \frac{1}{2}\left(x_{n-1} + \frac{a}{x_{n-1}^2}\right) \quad \text{oder} \quad x_n = \frac{1}{4}\left(x_{n-1} + \frac{3a}{x_{n-1}^2}\right) ,$$

indem Sie mit einem Programm hinreichend viele Glieder berechnen.

2.4. Werden zu Beginn eines jeden Jahres r DM auf ein Sparkonto gezahlt, so beträgt bei einem jährlichen Zinssatz p das Kapital Ende des n-ten Jahres

$$K_n = rq\,\frac{q^n - 1}{q - 1} \quad \text{mit} \quad q = 1 + p \text{ (Aufzinsungsfaktor).}$$

Berechnen Sie K_n mit Hilfe eines Programms für r = 1000.– DM für die folgenden Werte:

p	5,0; 5,25; 5,5; 5,75; 6,0; 6,25; 6,5 %
n	4; 5; 6; 7; 8 Jahre

2.5. Stellen Sie ein Programm auf zur Berechnung der reellen Lösungen der quadratischen Gleichung $a\,x^2 + b\,x + c = 0$.

Zahlenbeispiel:

a	1	3	0,52	1	9,41
b	−1	−7	2,71	−31,26	0,432
c	−6	2	0,86	112,4	−0,0316

(Die Lösungsformel lautet $x_{1,2} = \dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.)

2.6. Für die Auslenkung einer gedämpften Schwingung gilt

$y = A\,e^{-\delta t} \sin \omega t$.

Berechnen Sie für $A = 25$ mm, $\delta = 0{,}2\,\frac{1}{s}$ (Dämpfungsfaktor), $\omega = \frac{\pi}{2}\,\frac{1}{s}$ (Kreisfrequenz = Anzahl der Schwingungen in 2π Sekunden) und

$t = 0;\ 0{,}5;\ 1;\ 1{,}5;\ 2;\ \ldots\ 7{,}5;\ 8$ s

die Funktionswerte y. Zeichnen Sie die Schwingung in einem t, y-Koordinatensystem als Kurve.

2.7. Untersuchen Sie, was mit dem folgenden Programm berechnet wird.

PSS	Taste
000	STO
001	1
002	+
003	(
004	R/S
005	STO
006	2

007	x^2
008	+
009	1
010	)
011	$\sqrt{x}$
012	−
013	2
014	x

015	RCL
016	1
017	x
018	RCL
019	2
020	÷
021	(
022	RCL

023	1
024	+
025	RCL
026	2
027	)
028	=
029	R/S
030	RST

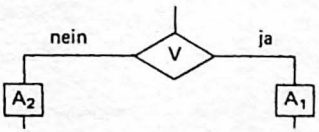
(Nennen Sie $(R_1) = a$ und $(R_2) = b$.)

3. Verzweigungen (bedingte Sprunganweisungen)

3.1. Das Flußdiagramm für Verzweigungen

Programmierbare Taschenrechner besitzen die Fähigkeit, einen **Vergleich** zwischen zwei Zahlenwerten durchzuführen. Sie können z.B. entscheiden, ob entweder $x \geqq 2$ oder $x < 2$ ist. Vom Ergebnis dieses Vergleichs wird die Fortsetzung des Rechenganges abhängen: *Wenn* $x \geqq 2$ ist, *dann* wird die Anweisung A_1 ausgeführt, *sonst* eine andere Anweisung A_2. Im Programmablauf können damit Sprünge durchgeführt werden, die von Bedingungen abhängen. Wir nennen sie daher **bedingte** Sprünge im Gegensatz zu den bereits früher kennengelernten *unbedingten* Sprüngen, die durch die Anweisungen GTO oder RST bewirkt werden.

Bevor wir auf die Programmierung bedingter Sprünge oder Verzweigungen eingehen, wollen wir vorher an zwei Beispielen das Flußdiagramm für solche Probleme entwickeln. Wir benutzen für einen Vergleich V oder für eine Verzweigung das Symbol



Fällt der Vergleich positiv aus, so wird die Anweisung A_1 ausgeführt, sonst A_2.

Beispiel: Von den in Bild 3.1.1 dargestellten Funktionen

$$y_1 = \frac{x^2}{4} + \sqrt{1+x} \quad \text{und} \quad y_2 = 4\,(1 - e^{-x})$$

sollen für einen gegebenen Wert $x \geqq -1$ die Funktionswerte berechnet werden. Von diesen beiden Werten y_1 und y_2 ist der größere – wir nennen ihn $y_{max} = \text{Max}\,(y_1; y_2)$ – herauszusuchen und anzuzeigen.

Zum Beispiel sind $y_1 = 1$ und $y_2 = 0$ für $x = 0$, also $y_{max} = \text{Max}\,(1; 0) = 1$.

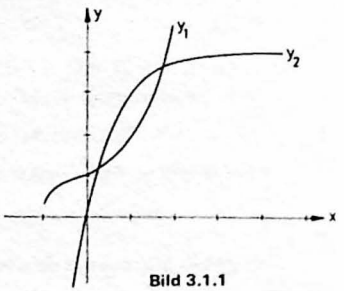
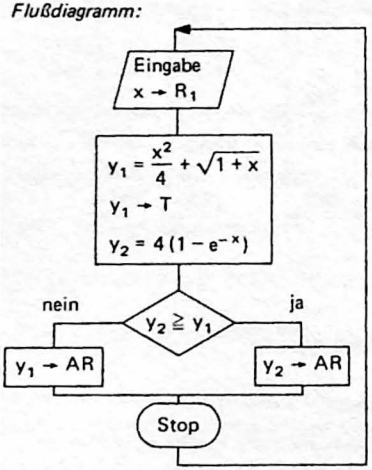


Bild 3.1.1

Flußdiagramm:



Die in diesem und dem folgenden Flußdiagramm benutzte Speicherung → T wird im nächsten Abschnitt erklärt.

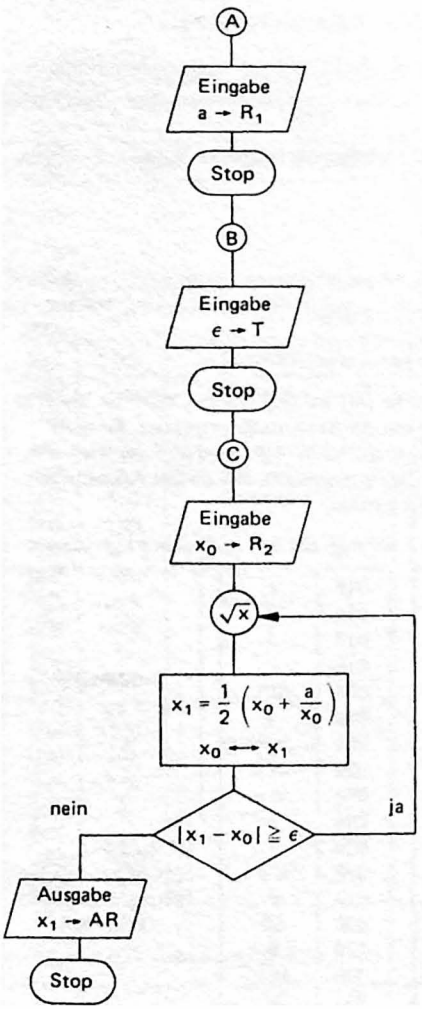
Beispiel: Quadratwurzelberechnung $x = \sqrt{a}$ durch Iteration:

$$x_n = \frac{1}{2}\left(x_{n-1} + \frac{a}{x_{n-1}}\right) \quad \text{für} \quad n \in \mathbb{N}\,.$$

Diese Aufgabe haben wir bereits früher (s. 1.3. und 2.4.) behandelt. Im Abschnitt 2.4. mußten wir nach jedem berechneten Wert x das Programm mit [B] manuell neu starten, um $\sqrt{a}$ ‚genau genug' berechnen zu können. Wir wollen jetzt erreichen, daß die Wiederholung der Rechnung selbständig durchgeführt wird und zwar solange, bis der neu berechnete Wert sich vom vorhergehenden um weniger als eine vorgegebene Größe ϵ (z.B. 0,000 001) unterscheidet. Es muß also iteriert werden, bis $|x_n - x_{n-1}| < \epsilon$ erfüllt ist.

Wir wollen im Programm Labels – für die Eingabe die Programmadreßtasten [A], [B], [C] und für die Verzweigung [$\sqrt{x}$] – benutzen.

Im *Flußdiagramm* setzen wir wie früher $x_{n-1} = x_0$ und $x_n = x_1$.



Der Austausch $x_0 \leftrightarrow x_1$ bedeutet $(AR) \rightarrow R_2 \wedge (R_2) \rightarrow AR$. Damit soll erreicht werden, daß x_1 im nächsten Iterationsschritt in *dem* Speicher vorgefunden wird, in dem vorher x_0 stand, *und* x_0 dabei nicht verloren geht, wie es früher (s. 2.4.) mit $x_0 := x_1$ der Fall war. x_0 benötigen wir hier noch für den Vergleich $|x_1 - x_0| \geqq \epsilon$.

3.2. Der T-Speicher (das T-Register)

Vergleiche zweier Zahlenwerte werden mit Hilfe eines besonderen Speichers (**T-Speicher** oder **T-Register**; T = Test) durchgeführt.

[x ⇆ t]	bewirkt einen Austausch der Inhalte des AR und des T-Speichers: (AR) ↔ (T)
[•CP]	als Anweisung in einem Programm löscht den T-Speicher.

[x ⇆ t] kann sowohl manuell in der Betriebsart RECHNEN als auch als Anweisung in einem Programm benutzt werden. Zur Erinnerung: [•CP] in der Betriebsart RECHNEN löscht alle Programmspeicher.

Mit den Zahlenwerten x = (AR) und t = (T) können die folgenden Vergleiche durchgeführt werden:

Ist $x = t$? Ist $x \neq t$?

Ist $x \geqq t$? Ist $x < t$?

Die entsprechenden Anweisungen lauten:

[•x = t]	n m p	[INV]	[•x = t]	n m p
[•x ≧ t]	n m p	[INV]	[•x ≧ t]	n m p

Anstelle der dreiziffrigen PSS kann eine Marke [M] benutzt werden. Wird der Vergleich positiv beantwortet, so wird das Programm mit *der* Anweisung fortgesetzt, die in der PSS n m p steht oder die auf [•LBL] [M] folgt. Andernfalls wird im Programm n m p oder [M] übergangen und die nächste Anweisung ausgeführt. Die obigen Anweisungen dürfen in unvollständige Operationen gesetzt werden.

Beispiel: Wir stellen das Programm für das 1. Beispiel aus 3.1. nach dem Flußdiagramm auf

PSS	Taste	Bemerkungen
000	STO	$x \rightarrow R_1$
001	1	
002	+	Berechnung von y_1
003	1	
004	=	
005	$\sqrt{x}$	
006	+	
007	RCL	
008	1	
009	x^2	
010	÷	
011	4	
012	=	
013	x ⇆ t	$y_1 \rightarrow T$
014	4	
015	x	Berechnung von y_2
016	(	
017	1	
018	−	
019	RCL	
020	1	
021	+/−	
022	INV	
023	ln x	
024	)	
025	=	Ist $y_2 \geqq y_1$?
026	•x ≧ t	Sprung auf PSS 030,
027	0	falls $y_2 \geqq y_1$, sonst
028	30	$y_1 = (T) \rightarrow AR$
029	x ⇆ t	
030	R/S	
031	RST	

Benutzeranleitung:

Maximum zweier Funktionswerte y_1 und y_2				
Programm eintasten.				
Speicherplan		Eingabe	Taste	Anzeige
T	y	x	R/S	y_{max}
1	x	x	R/S	y_{max}
		...	usw.	...

Zahlenbeispiel (Angabe auf 3 Nachkommastellen):

x	0	−1	1	2	3	4
y_{max}	1,000	0,250	2,528	3,459	4,250	6,236

Beispiel: Das Programm zur Berechnung der Quadratwurzel (2. Beispiel aus 3.1) auf eine vorgegebene Genauigkeit ϵ sieht folgendermaßen aus:

PSS	Taste	Bemerkungen
000	*LBL	Eingabe a
001	A	
002	STO	$a \rightarrow R_1$
003	1	
004	R/S	Eingabe ϵ
005	*LBL	
006	B	
007	x ▶ t	$\epsilon \rightarrow T$
008	R/S	Eingabe x_0
009	*LBL	
010	C	
011	STO	$x_0 \rightarrow R_2$
012	2	
013	*LBL	Marke $\sqrt{x}$
014	$\sqrt{x}$	
015	RCL	
016	2	
017	+	
018	RCL	
019	1	
020	÷	Berechnung von x_1
021	RCL	
022	2	
023	=	
024	÷	
025	2	
026	=	
027	*Pause	
028	*Exc	$x_1 \rightarrow R_2 \wedge x_0 \rightarrow AR$
029	2	
030	−	
031	RCL	Berechnung von $\lvert x_1 - x_0 \rvert$
032	2	
033	=	
034	*|x|	
035	*x ≧ t	$\lvert x_1 - x_0 \rvert \geq \epsilon$?
036	$\sqrt{x}$	Sprung auf PSS 015, falls $\lvert x_1 - x_0 \rvert \geq \epsilon$, sonst $\sqrt{a} \rightarrow AR$
037	RCL	
038	2	
039	R/S	
040		

In der PSS 027 haben wir die Anweisung [*Pause] benutzt. Der Rechner zeigt dann einen kurzen Augenblick den im AR stehenden Zahlenwert an. Auf diese Weise können wir die Zahlenwerte der Iterationsfolge $x_1, x_2, x_3, \ldots$ *‚beobachten'*.

Benutzeranleitung:

Quadratwurzelberechnung $x = \sqrt{a}$				
1. Programm eintasten. 2. Die Iteration wird abgebrochen, wenn $\lvert x_n - x_{n-1}\rvert < \epsilon$ geworden ist.				
Speicherplan		Eingabe	Taste	Anzeige
T	ϵ	a	A	–
1	a	ϵ	B	–
2	x_0	x_0	C	$\sqrt{a}$

Führen Sie selbst mit diesem Programm die Berechnung von $\sqrt{14{,}256}$ für verschiedene Ausgangswerte x_0 und Genauigkeiten ϵ durch. Berechnen Sie auch andere Quadratwurzeln mit dem gespeicherten Programm.

3.3. Mehrfache Verzweigungen

Wie Programme mit mehr als einer Verzweigung aufgebaut werden, zeigen wir am besten an zwei Beispielen. Der Leser wird schnell erkennen: Je genauer ein Flußdiagramm erstellt wird, um so leichter und sicherer läßt sich das Programm schreiben. Besonders der Anfänger im Programmieren sollte großen Wert auf ein ausführliches und exaktes Flußdiagramm legen. Mit zunehmender Programmiererfahrung wird dieses ganz von allein immer spärlicher ausfallen.

Beispiel: Eine Bank bietet ihren Sparern für ein eingezahltes Kapital K_0:

im 1. Jahr $p_1 = 4{,}25\,\%$ Zinsen,
im 2. Jahr $p_2 = 5{,}00\,\%$ Zinsen,
im 3. Jahr $p_3 = 6{,}50\,\%$ Zinsen,
im 4. Jahr $p_4 = 7{,}50\,\%$ Zinsen,
im 5. Jahr $p_5 = 8{,}50\,\%$ Zinsen.

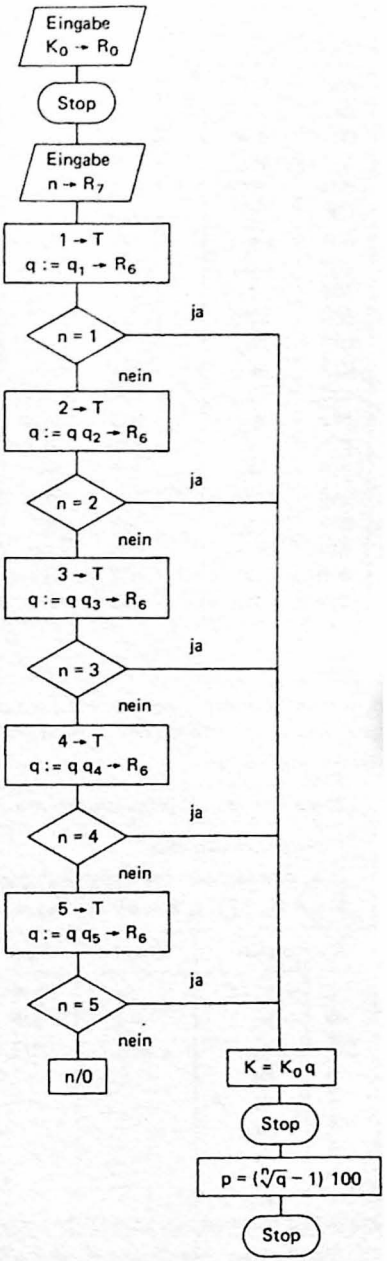
Mit einem Programm sollen für K_0 nach $n \in \mathbb{N}_5$ Jahren das Kapital K_n und der mittlere Zinssatz p (die Rendite) ausgerechnet werden. Bei einer Eingabe eines nicht zulässigen n-Wertes ($n \notin \mathbb{N}_5$) soll dieses durch Blinken (z.B. Division durch Null) angezeigt werden.

Es gilt: $K_n = K_{n-1}(1 + p_n) = K_{n-1}q_n = K_{n-2}q_{n-1}q_n = \ldots = K_0 q_1 q_2 \ldots q_n$.

Wir setzen $q := q_1$, $q := q\ q_2$, ..., $q := q\ q_n$ und berechnen $K_n = K_0 q$ und $p = (\sqrt[n]{q} - 1)\ 100$.

Die Aufzinsungsfaktoren q_n speichern wir manuell: $q_n \rightarrow R_n$.

Flußdiagramm
(Speicherplan s. Benutzeranleitung):



Das Programm lautet:

PSS	Taste
000	STO
001	0
002	R/S
003	STO
004	07
005	1
006	x ⇄ t
007	RCL
008	1
009	STO
010	6
011	RCL
012	7
013	*x = t
014	0
015	63
016	2
017	x ⇄ t
018	RCL
019	2
020	*Prd

021	6
022	RCL
023	7
024	*x = t
025	0
026	63
027	3
028	x ⇄ t
029	RCL
030	3
031	*Prd
032	6
033	RCL
034	7
035	*x = t
036	0
037	63
038	4
039	x ⇄ t
040	RCL
041	4
042	*Prd

043	6
044	RCL
045	7
046	*x = t
047	0
048	63
049	5
050	x ⇄ t
051	RCL
052	5
053	*Prd
054	6
055	RCL
056	7
057	*x = t
058	0
059	63
060	÷
061	0
062	=
063	RCL
064	0

065	×
066	RCL
067	6
068	=
069	R/S
070	RCL
071	6
072	INV
073	y^x
074	RCL
075	7
076	−
077	1
078	=
079	×
080	1
081	0
082	0
083	=
084	R/S
085	RST

Benutzeranleitung:

Anwachsen eines Kapitals und Rendite

1. Programm eintasten.
2. q_n manuell nach Speicherplan eingeben.
3. $n \in \mathbb{N}_5 = \{1, 2, 3, 4, 5\}$; $n \notin \mathbb{N}_5$ erzeugt Blinken.

Speicherplan		Eingabe	Taste	Anzeige
T	1, 2 ... 5	K_0	R/S	−
0	K_0	n	R/S	K_n
1	q_1	−	R/S	p
2	q_2			
3	q_3			
4	q_4			
5	q_5			
6	q			
7	n			

Zahlenbeispiel (mit [*Fix] 2):

n↓	K_0 →	1 000,–	3 500,–	DM
1	K_n	1 042,50	3 648,75	DM
	p	4,25	4,25	%
4	K_n	1 253,21	4 386,23	DM
	p	5,80	5,80	%
5	K_n	1 359,73	4 759,06	DM
	p	6,34	6,34	%
6	K	*Blinken*		
	p			

Beispiel: Für $x \in \{0;\ 0{,}5;\ 1;\ 1{,}5;\ 2;\ 2{,}5;\ 3\}$ sollen die Werte der Funktion

$$y = \begin{cases} 0{,}4 + \frac{x}{2} & \text{für } 0 \leq x < 1 \\ 2{,}5\, e^{\frac{x-1}{3}} & \text{für } 1 \leq x < 2 \\ \frac{15}{1+x^2} & \text{für } 2 \leq x < \infty, \end{cases}$$

deren Kurve in Bild 3.3.1 gezeichnet ist, und $z = \sqrt{1+y^2}$ berechnet werden. Zum Schluß einer Rechnung sollen nacheinander die zugehörigen x-, y- und z-Werte vom Rechner angezeigt werden.

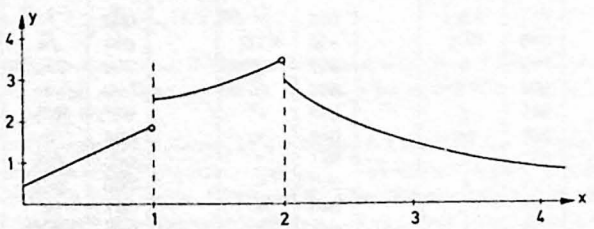
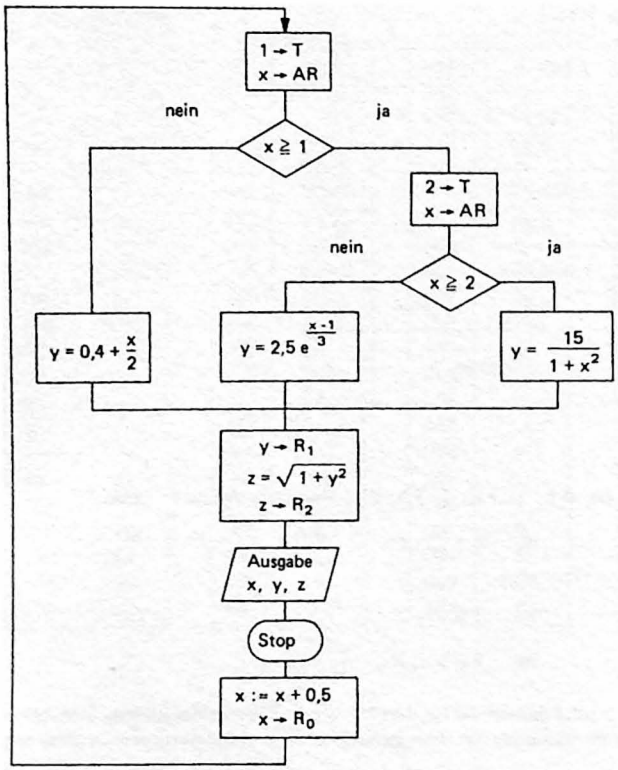


Bild 3.3.1

Im *Flußdiagramm* fassen wir die Ausgabe der x-, y- und z-Werte für alle drei Veränderliche zusammen. Da wir die Rechnung für äquidistante x-Werte durchführen, benutzen wir $x := x + 0{,}5$ (s. 2.2.).



In dem folgenden Programm sind die Verzweigungen erkennbar an den Anweisungen [•x ≧ t].
Die gemeinsame Rechnung (im Flußdiagramm ab y → R_1) wird ab PSS 048 durchgeführt.

PSS	Taste
000	1
001	x ⇆ t
002	RCL
003	0
004	•x ≧ t
005	0
006	16
007	÷
008	2
009	+
010	.
011	4
012	=
013	GTO
014	0
015	48
016	2

017	x ⇆ t
018	RCL
019	0
020	•x ≧ t
021	0
022	39
023	–
024	1
025	=
026	÷
027	3
028	=
029	INV
030	ln x
031	x
032	2
033	.
034	5

035	=
036	GTO
037	0
038	48
039	x^2
040	+
041	1
042	=
043	÷
044	1
045	5
046	=
047	1/x
048	STO
049	1
050	x^2
051	+
052	1

053	=
054	$\sqrt{x}$
055	STO
056	2
057	RCL
058	0
059	R/S
060	RCL
061	1
062	R/S
063	RCL
064	2
065	R/S
066	.
067	5
068	SUM
069	0
070	RST

Benutzeranleitung:

Berechnung von Funktionswerten				
1. Programm eintasten. 2. Taste [*CM$_s$] betätigen.				
Speicherplan		Eingabe	Taste	Anzeige
0	x	–	R/S	x
1	y	–	R/S	y
2	z	–	R/S	z
T	1; 2	–	R/S	x
		...	usw.	...

Zahlenbeispiel (mit [*Fix] 3):

x	0	0,5	1	1,5	2	2,5	3
y	0,400	0,650	2,500	2,953	3,000	2,069	1,500
z	1,077	1,193	2,693	3,118	3,162	2,298	1,803

3.4. Verminderung und Überspringen bei Null

Neben [*x = t], [*x ≧ t] usw. (s. 3.2.) gibt es eine weitere Verzweigungsanweisung, die insbesondere bei Berechnungen von Summen oder Folgen mit Vorteil benutzt werden kann. Wir wollen sie in den nächsten Beispielen erläutern und anwenden.

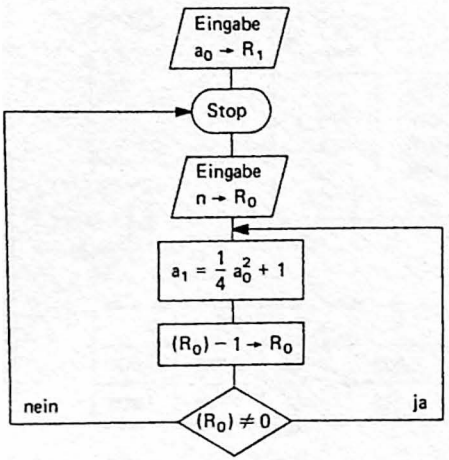
Beispiel: Die Zahlenfolge

$$a_n = 1 + \frac{1}{4} a_{n-1}^2 \qquad (n \in \mathbb{N})$$

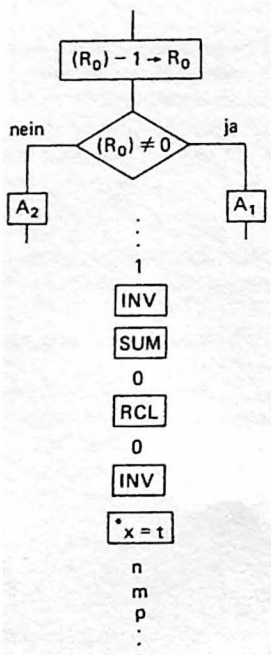
soll für verschiedene Ausgangswerte a_0 untersucht werden. Mit einem Programm wollen wir einige a_n für verschiedene a_0 und n berechnen, z.B. $a_{10}, a_{50}, a_{100}, a_{1000}$ für $a_0 = 1$ oder $a_0 = 1{,}5$.

Wir nennen eine Zahlenfolge der obigen Art *rekursiv:* Das Folgenglied wird aus den vorhergehenden Folgengliedern – hier a_{n-1} – berechnet. Dagegen ist z.B. $a_n = n(1 + n^2)$ keine rekursive Zahlenfolge. Hier kann a_n sofort für jede Zahl $n \in \mathbb{N}$ *ohne* Kenntnis der vorhergehenden Folgenglieder ausgerechnet werden. Rekursive Zahlenfolgen haben wir bereits früher bei der iterativen Berechnung der Quadratwurzel (s. 2.4.) oder der Summe der Quadratzahlen (s. 2.2.) oder der Funktionswerte einer quadratischen Funktion (s. 2.4.) kennengelernt.

Bei der Berechnung des Folgengliedes a_n für ein gegebenes n müssen wir dafür sorgen, daß die Rechnung gestoppt wird, sobald n erreicht ist. Andernfalls muß nach der obigen Rekursionsformel das nächste Folgenglied berechnet werden. Setzen wir $a_{n-1} = a_0$ und $a_n = a_1$, so kann das *Flußdiagramm* folgendermaßen aussehen:



Sehen wir uns im Flußdiagramm die Verzweigung und die vorausgehende Anweisung noch einmal genauer an.



Der Inhalt des Speichers R_0 wird um 1 vermindert. Es stehen also der Reihe nach in R_0 die Zahlen:

n, n − 1, n − 2, ..., 3, 2, 1, 0.

Ist der Inhalt von R_0 *nicht* Null, so findet eine Verzweigung nach der PSS n m p statt, in der die Anweisung A_1 steht. Andernfalls wird das Programm mit der Anweisung A_2 fortgesetzt.

Die Programmfolge für diesen Ausschnitt ist nebenstehend (mit (T) = 0) angegeben.

Die ersten acht Programmschritte können vom Rechner mit *einer* Anweisung erfaßt werden.

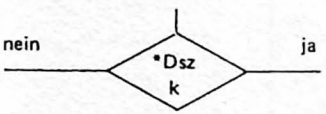
Die Anweisung [*Dsz] k n m p oder [*Dsz] k [M] mit $k \in \mathbb{N}_{0,9}$ (Decrement and Skip on Zero) bewirkt:

1. Verminderung des Inhalts des Speichers R_k um 1.
2. Ist der Inhalt von R_k ungleich Null, so findet eine Verzweigung nach der PSS n m p oder der auf [*LBL] [M] folgenden PSS statt, sonst wird n m p oder [M] übergangen und die darauffolgende Anweisung ausgeführt.

Mit [INV] [*Dsz] k n m p wird entsprechend für $(R_k) = 0$ eine Verzweigung auf die PSS n m p erreicht.

[*Dsz] oder [INV] [*Dsz] darf in eine unvollständige Operation gesetzt werden.

Wir werden bei den folgenden Aufgaben stets



statt des obigen Flußdiagrammausschnitts benutzen.

Das Programm für unser Beispiel lautet:

PSS	Taste	Bemerkungen
000	STO	$a_0 \rightarrow R_1$
001	1	
002	R/S	Eingabe n
003	STO	$n \rightarrow R_0$
004	0	
005	RCL	$a_0 \rightarrow$ AR
006	1	
007	x^2	
008	÷	
009	4	Berechnung von a_1
010	+	
011	1	
012	=	
013	*Dsz	$(R_0) - 1 \rightarrow R_0 \wedge (R_0) \neq 0$?
014	0	Sprung auf PSS 007, falls
015	0	$(R_0) \neq 0$, sonst
016	07	Sprung auf
017	GTO	PSS 002
018	0	
019	02	

Benutzeranleitung:

Berechnung einer Folge

1. Programm eintasten.
2. Vor der Eingabe eines neuen Wertes a_0 muß das BAR mit [RST] auf 000 gestellt werden.

Speicherplan	Eingabe	Taste	Anzeige
0 n	a_0	R/S	–
1 a_0	n	R/S	a_n
	n	R/S	a_n
	n	R/S	a_n
	...	usw.	...

In der folgenden Tabelle sind die Werte a_n für einige a_0 und n angegeben[1]. Bei dieser Aufgabe haben wir außerdem die Zeit gestoppt, die der Rechner von der Eingabe von n und dem Betätigen der Taste R/S bis zur Anzeige des Ergebnisses benötigt. Bei der Iterationsrechnung werden in einer Rechenschleife 3 Rechenoperationen durchgeführt, d.h. wir kommen auf etwa 7 Rechenoperationen pro Sekunde. Auf etwa diese Zahl kommt man bei anderen Tests ebenfalls. Natürlich wird für das Speichern und die Sprunganweisung auch noch Zeit benötigt, so daß die Anzahl der Rechenoperationen pro Sekunde größer als 7 ist.

a_0	n	a_n	Rechenzeit [s]
1	10	1,741490131	5
	50	1,929575725	23
	100	1,962770003	45
	1000	1,996038367	446
1,5	10	1,788369589	5
	50	1,933418015	23
	100	1,963863469	45
	1000	1,996050978	447
	2000	1,998013503	900
2	10	2	4,8
	50	2	20,5
	100	2	39,5

Bemerkung: Führen Sie mit dem obigen Programm die Rechnung für $a_0 = 1$ und $n = 9{,}5$ durch. Sie kommen auf den oben angegebenen Wert a_{10}. Geben Sie auch einmal ein: $n = 0{,}5$. oder $n = -0{,}5$ oder $n = -10$ oder $n = -9{,}1$. Bei diesen n-Werten sollten wir eigentlich eine Fehlermeldung durch Blinken erwarten! Tatsächlich arbeitet die Anweisung *Dsz für nicht-natürliche Zahlen so: Bei *Dsz k ersetzt der Rechner intern den Inhalt des Speichers R_k durch die kleinste natürliche Zahl, die größer oder gleich $|(R_k)|$ ist. Für diese Zahl wird die Verminderung um 1 und Sprung bei $(R_k) \neq 0$ durchgeführt.

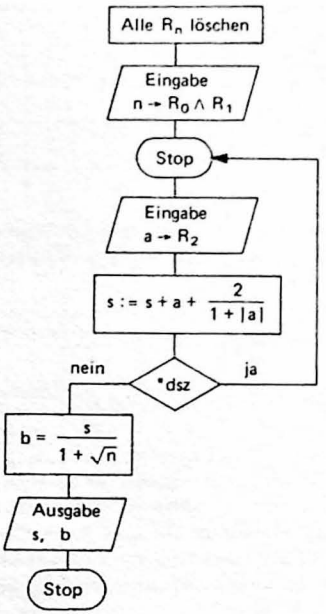
Beispiel: Berechnung von

$$s = \sum_{k=1}^{n} \left(a_k + \frac{2}{1 + |a_k|} \right) \quad \text{und} \quad b = \frac{s}{1 + \sqrt{n}}$$

für gegebene Werte n und a_k $(k \in \mathbb{N}_n)$. Nach der letzten eingegebenen Zahl sollen s und b nacheinander vom Rechner ausgegeben werden.

[1] Für mathematisch Interessierte: Die Zahlenfolge (a_n) ist konvergent mit dem Grenzwert 2 für $|a_0| \leqq 2$, dagegen divergent für $|a_0| > 2$. Geben Sie z.B. $a_0 = 2{,}1$ oder $a_0 = 3$ ein und beobachten Sie, was der Rechner macht.

Flußdiagramm:



Programm:

PSS	Taste
00	$*CM_s$
01	STO
02	0
03	STO
04	1
05	0
06	R/S
07	STO
08	2

09	+
10	2
11	÷
12	(
13	1
14	+
15	RCL
16	2
17	*|x|
18	)

19	=
20	SUM
21	3
22	*dsz
23	0
24	5
25	RCL
26	3
27	R/S
28	÷

29	(
30	1
31	+
32	RCL
33	1
34	$*\sqrt{x}$
35	)
36	=
37	R/S
38	RST

Benutzeranleitung:

Berechnung einer Summe

1. Programm eintasten.
2. Vor der Eingabe des nächsten a-Wertes erscheint im AR eine Null.

Speicherplan		Eingabe	Taste	Anzeige
0	n, ..., 0	n	R/S	0
1	n	a_1	R/S	0
2	a	a_2	R/S	0
3	s	...	...	...
		a_n	R/S	s
		–	R/S	b

Um das Programm zu testen, setzen wir $a_k = 1$ und z.B. $n = 9$. Dann wird $s = 2n = 18$ und $b = \frac{2n}{1+\sqrt{n}} = 4{,}5$.

Zahlenbeispiel: $n = 7$

a_k	1,5	− 3,2	4,6	5,2	2,2	− 1,9	1

Ergebnis: $s = 13{,}671$ und $b = 3{,}750$.

In dem folgenden Beispiel treten neben Verzweigungs- auch ‚Umordnungsprobleme' auf, die erfahrungsgemäß dem Anfänger häufig Schwierigkeiten bereiten.

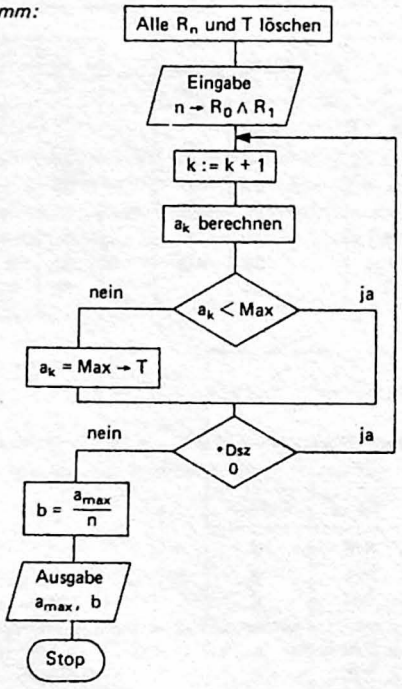
Beispiel: Von den n (positiven) Zahlen

$$a_k = \sqrt{k} + \sin\frac{\pi k}{n} \quad (k \in \mathbb{N}_n)$$

sollen die größte Zahl $a_{max} = \text{Max}(a_1, a_2, \ldots, a_n)$ und $b = \frac{a_{max}}{n}$ berechnet und nacheinander angezeigt werden.

Nehmen wir einmal an, wir haben von den Zahlen $a_1, a_2, \ldots, a_{k-1}$ die größte – kurz Max genannt – bereits im T-Speicher stehen. Dann müssen wir die nächste Zahl a_k mit Max vergleichen und *nicht* austauschen, falls $a_k < \text{Max}$ ist. Andernfalls nehmen wir einen Austausch vor und führen den nächsten Vergleich mit a_{k+1} durch. Dieses führen wir solange durch, bis der letzte Wert a_n mit dem bis dahin größten Wert verglichen wurde. Danach bringen wir $a_{max} = (T)$ ins AR zur Anzeige. Wir erhalten somit das folgende

Flußdiagramm:



Das Programm lautet:

PSS	Taste
000	*CM$_s$
001	*CP
002	STO
003	0
004	STO
005	01
006	1
007	SUM
008	2
009	RCL

010	2
011	×
012	*π
013	÷
014	RCL
015	1
016	=
017	*sin
018	+
019	RCL
020	2

021	$\sqrt{x}$
022	=
023	INV
024	*x ≧ t
025	0
026	28
027	x ↹ t
028	*Dsz
029	0
030	0
031	06

032	x ↹ t
033	R/S
034	÷
035	RCL
036	1
037	=
038	R/S
039	RST

Benutzeranleitung:

Größtes Folgenglied einer Zahlenfolge

1. Programm eintasten.
2. Mit [*Rad] Bogenmaß einstellen.

Speicherplan		Eingabe	Taste	Anzeige
T	a_{max}	n	R/S	a_{max}
0	n, ..., 0	–	R/S	b
1	n	n	R/S	a_{max}
2	k	–	R/S	b
		...	usw.	...

Zahlenbeispiel (mit [*Fix] 3)[1]:

n	5	10	20	30	40	50	60
a_{max}	2,683	3,455	4,588	5,506	6,325	7,071	7,746
b	0,537	0,345	0,229	0,184	0,158	0,141	0,129

3.5. Flags (Flaggen, Signale)

Bei den bisherigen Verzweigungen wurden in einem Vergleich – z.B. $x = t$, $x < \tau$, $(R_k) \neq 0$ o.ä. – zwei Zahlenwerte miteinander verglichen. Vom Ergebnis dieses Vergleichs hing der weitere Programmablauf ab. Eine Verzweigung, die wir uns hier anschaulich als eine „Weichenstellung" vorstellen können, kann mit den TI-58/59 – Rechnern auch ohne einen numerischen Vergleich durchgeführt werden.

Beispiel: Wir wollen ein Programm zur Berechnung von Funktionswerten y_1 oder y_2 schreiben:

$$y_1 = \left(\frac{x^2}{3} + 4x + 5 + \sqrt{1+x^2}\right) e^{-x/2} \qquad \text{für } x \in M_1;$$

$$y_2 = \left(\frac{x^2}{3} + 4x + 5 + \sqrt{1+x^2}\right) \ln(1 + |x|) \qquad \text{für } x \in M_2.$$

Hierin bedeuten M_1 und M_2 reelle Zahlenmengen mit $M_1 \cap M_2 = \emptyset$,
z.B. $M_1 = \{-0{,}5;\ 1{,}2;\ 2{,}3;\ 2{,}9\}$ und $M_2 = \{-1{,}1;\ 0;\ 0{,}8;\ 1{,}5;\ 3{,}1\}$.
Für einen gegebenen Wert $x \in M_1 \cup M_2$ läßt sich durch einen numerischen Vergleich – z.B. $x \geq 1$ o.ä. – für die obigen Mengen M_1 und M_2 keine Verzweigung zur Berechnung von y_1 oder y_2 durchführen. Man könnte natürlich für jede Funktion ein eigenes Programm schreiben, das z.B. mit einer Programmadreßtaste [A] oder [B] aufgerufen wird.

[1] Für mathematisch Interessierte: Für $n \geq 40$ erhält man stets $a_{max} = \sqrt{n}$ und $b = \frac{1}{\sqrt{n}}$.

Dann wird aber für den gemeinsamen Faktor

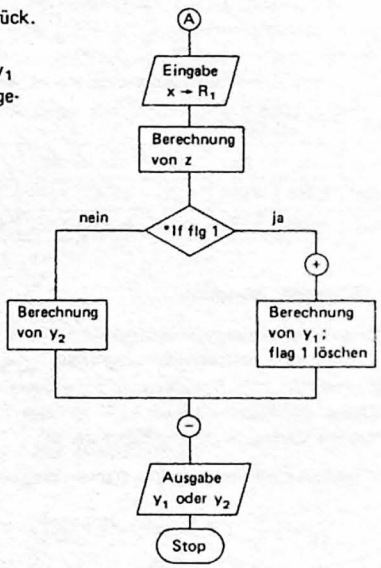
$$z = \frac{x^2}{3} + 4x + 5 + \sqrt{1 + x^2}$$

zweimal eine Anweisungsfolge geschrieben. Dieses wollen wir vermeiden. Der Rechner muß nach der Berechnung von z einen Hinweis erhalten, ob er diesen Wert entweder mit $e^{-x/2}$ oder mit $\ln(1 + |x|)$ zu multiplizieren hat. Diesen Hinweis geben wir dem Rechner durch das Setzen von **flags** (Flaggen, Signale). Ist ein flag gesetzt worden, so wird der Rechner z.B. y_1 berechnen, andernfalls y_2.

Durch [•St flg] k (Set flag) für $k \in \mathbb{N}_{0,9}$ können bis zu 10 flags gesetzt werden. [•St flg] k kann manuell in der Betriebsart RECHNEN oder als Anweisung in einem Programm benutzt werden. Mit [INV] [•St flg] k wird die Markierung durch das flag k zurückgenommen, mit [RST] (s. 2.1.) werden alle 10 flags gelöscht.

Die Anweisung [•If flg] k n m p oder [•If flg] k [M] bewirkt einen Sprung auf die PSS n m p oder auf die auf [•LBL] [M] folgende PSS, falls flag k gesetzt wurde. Andernfalls wird n m p oder [M] übergangen und die nächste Anweisung im Programm ausgeführt. – Entsprechendes gilt für [INV] [•If flg] k.

Kehren wir zu unserem Beispiel zurück. Zunächst zeichnen wir ein *Flußdiagramm*. Wir setzen das flag 1, falls y_1 berechnet werden soll. Ist es nicht gesetzt, so wird y_2 berechnet.



Sehen wir uns zunächst einmal die Marken an. Die Programmadreßtaste [A] startet das Programm nach dem Eintasten des x-Wertes. Die Marke (+) markiert die Stelle im Programm, von der ab nach positivem Test (If flg 1?) die nächste Anweisung ausgeführt werden soll. Die Marke (−) leitet die Ausgabe des berechneten Wertes y_1 oder y_2 ein. Während wir die Marken (+) oder (−) durch beliebige andere ersetzen dürfen (z.B. (•sin) oder (1/x) oder (E)), werden wir für das manuelle Starten des Programms stets eine Programmadreßtaste benutzen (es geht natürlich umständlicher auch mit [GTO] [M] [R/S]).

Das Setzen von flag 1 für $x \in M_1$ wird hier manuell vor dem Starten des Programms durchgeführt. Gelöscht wird flag 1 im Programm durch die Anweisung in den PSS 052 bis 054.

Das Programm lautet:

PSS	Taste
000	*LBL
001	A
002	STO
003	1
004	x^2
005	÷
006	3
007	+
008	4
009	x
010	RCL
011	1
012	+
013	5

014	+
015	(
016	1
017	+
018	RCL
019	1
020	x^2
021	)
022	$\sqrt{x}$
023	=
024	*If flg
025	1
026	+
027	x
028	(

029	1
030	+
031	RCL
032	1
033	*|x|
034	)
035	ln x
036	=
037	GTO
038	–
039	*LBL
040	+
041	x
042	(
043	RCL

044	1
045	+/–
046	÷
047	2
048	)
049	INV
050	ln x
051	=
052	INV
053	*St flg
054	1
055	*LBL
056	–
057	R/S
058	

Benutzeranleitung:

Berechnung von Funktionswerten y_1 für $x \in M_1$ oder y_2 für $x \in M_2$.				
Für $x \in M_1$	*St flg	1 betätigen.		
Speicherplan		Eingabe	Taste	Anzeige
1	x	x	A	y

Wir testen das Programm mit $x = 0 \Rightarrow y_1 = 6$; $y_2 = 0$. Danach berechnen wir für die (geordneten) Werte $x \in M_1 \cup M_2$ die Funktionswerte y auf 2 Nachkommastellen:

x	–1,1	–0,5	0	0,8	1,2	1,5	2,3	2,9	3,1
y	1,85	5,39	0,00	5,70	6,50	12,42	5,85	5,27	33,67

Beispiel: Für die Zahlenwerte a_k ($k \in \mathbb{N}_n \wedge n$ ist Vielfaches von 3) sollen die Summe

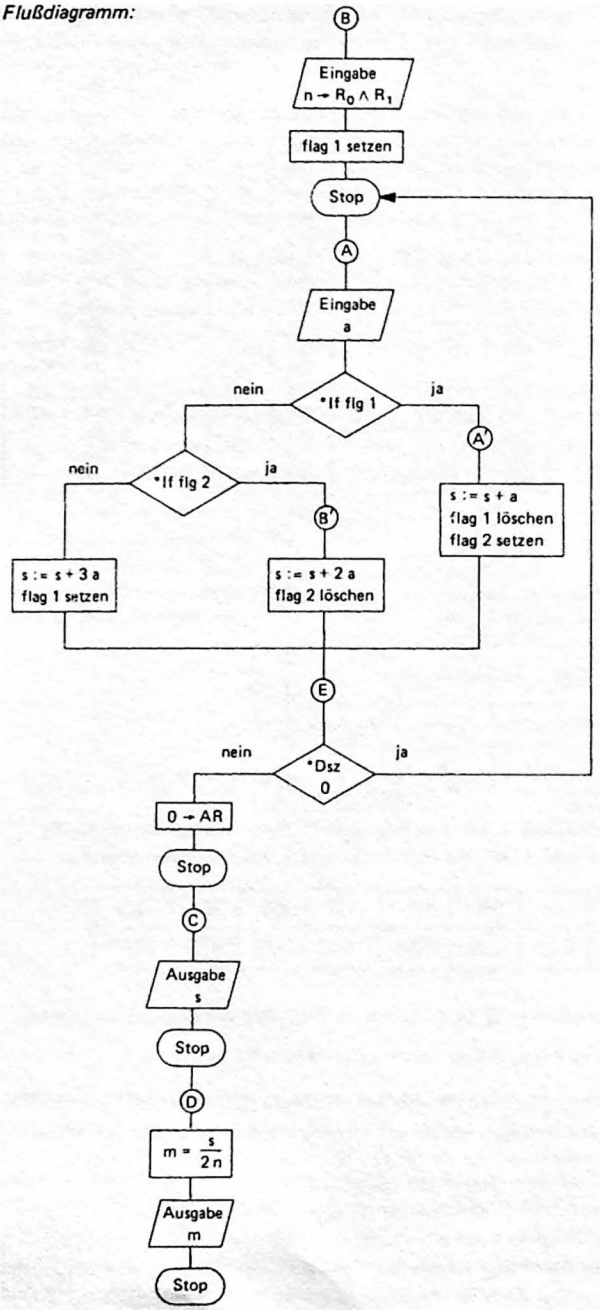
$s = a_1 + 2\,a_2 + 3\,a_3 + a_4 + 2\,a_5 + 3\,a_6 + \ldots + a_{n-2} + 2\,a_{n-1} + 3\,a_n$

und der Mittelwert $m = \frac{s}{2n}$ berechnet werden. n und a_k werden manuell eingegeben.
Bei der Berechnung von s müssen wir bei der Eingabe von a unterscheiden zwischen

$a_1, a_4, a_7, \ldots \Rightarrow s := s + a$;
$a_2, a_5, a_8, \ldots \Rightarrow s := s + 2\,a$;
$a_3, a_6, a_9, \ldots \Rightarrow s := s + 3\,a$.

Den 1. Fall wollen wir durch flag 1 markieren, den 2. durch flag 2, während im 3. keine flags gesetzt werden. – Das Ende der Berechnung soll durch eine 0 im AR angezeigt werden.

Flußdiagramm:



Verfolgen Sie beim Schreiben des Programms genau die einzelnen Wege im Flußdiagramm und die zugehörigen Anweisungsfolgen im Programm.

PSS	Taste
000	*LBL
001	B
002	STO
003	0
004	STO
005	1
006	*St flg
007	1
008	R/S
009	*LBL
010	A
011	*If flg
012	1
013	*A'
014	*If flg
015	2
016	*B'

017	×
018	3
019	=
020	SUM
021	2
022	*St flg
023	1
024	E
025	*LBL
026	*A'
027	SUM
028	2
029	INV
030	*St flg
031	1
032	*St flg
033	2
034	E

035	*LBL
036	*B'
037	×
038	2
039	=
040	SUM
041	2
042	INV
043	*St flg
044	2
045	*LBL
046	E
047	*Dsz
048	0
049	0
050	08
051	0
052	R/S

053	*LBL
054	C
055	RCL
056	2
057	R/S
058	*LBL
059	D
060	RCL
061	2
062	÷
063	2
064	÷
065	RCL
066	1
067	=
068	R/S
069	
070	

Eingabe: $\boxed{CM_s}$ n $\boxed{B}$ a_1 $\boxed{A}$ a_2 $\boxed{A}$ ... a_n $\boxed{A}$

Ausgabe: $\boxed{C}$ s $\boxed{D}$ m.

Wir führen für dieses Programm zwei Tests durch:

1. $n = 3;\ a_k = 1 \Rightarrow s = 6;\quad m = 1;$
2. $n = 6;\ a_k = k \Rightarrow s = 46;\ m = 3{,}83333333.$

Nach einem erfolgreichem Test können wir für beliebige n und a_k die Summe s und den Mittelwert m berechnen.

3.6. Übungsaufgaben

3.1. Berechnen Sie für $x \in \{0{,}5;\ 4;\ 1;\ 2;\ 0;\ 3\}$ die Werte der Funktion

$$y = \begin{cases} \dfrac{x^2}{4} & \text{für } 0 \leq x \leq 2; \\ \dfrac{2x-1}{x+1} & \text{für } 2 \leq x. \end{cases}$$

(*Zusatz:* Die Eingabe eines negativen (nicht zulässigen) x-Wertes soll der Rechner durch Blinken anzeigen.)

3.2. Berechnen Sie

$$s = \sum_{k=1}^{n} (a_k + c)\, b_k^2; \quad r = \sqrt{c^2 + \sqrt{s}}; \quad p = \frac{c\,s}{1+n}$$

für $n = 6;\ c = 1{,}9$ und

a_k	2,9	7,3	− 4,7	0,87	− 3,6	2,9
b_k	5,1	− 4,9	6,3	1,23	4,1	− 5,1

3.3. Von der Folge

$$a_n = \frac{3}{a_{n-1}} + \frac{\sqrt{a_{n-1}}}{n+1} \qquad (n \in \mathbb{N})$$

sind die Folgenglieder a_5, a_{20}, a_{100} für $a_0 = 1, 2, 10$ zu berechnen.

3.4. Berechnen Sie die Zahl π aus den Reihendarstellungen

a) $$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \ldots + \frac{(-1)^{n+1}}{2n-1} + \ldots$$

b) $$\frac{\pi}{6} = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{3 \cdot 2^3} + \frac{1 \cdot 3}{2 \cdot 4} \cdot \frac{1}{5 \cdot 2^5} + \ldots + \frac{1 \cdot 3 \cdot \ldots \cdot (2n-1)}{2 \cdot 4 \cdot \ldots \cdot (2n)} \cdot \frac{1}{(2n+1) \cdot 2^{2n+1}} + \ldots$$

Der Rechner soll die Rechnung solange fortsetzen, bis der neu berechnete Summenwert sich vom vorhergehenden um weniger als ϵ unterscheidet. Wählen Sie z.B. für a) $\epsilon = 0{,}01$ und für b) $\epsilon = 0{,}000001$. (Von den obigen Reihen konvergiert die erste sehr schlecht. Sie ist aber einfach zu programmieren, während die zweite Reihe schon einige Programmierfähigkeit voraussetzt. Versuchen Sie hier, für den Summenwert und für das Reihenglied Rekursionsformeln aufzustellen.)

3.5. Berechnen Sie für $m \in \mathbb{N}_n$ die Summenwerte

$$s_m = \sum_{k=1}^{m} \frac{2{,}1 + b_k^2}{0{,}8 + b_k}$$ für $b_k \leqq a$ und den Zahlenwert $c = \frac{a + \sqrt{s_n}}{n}$.

Zahlenbeispiel: $n = 6$; $a = 3{,}5$;

b_k	4,72	2,83	1,46	5,90	3,50	3,14

(*Hinweis:* $\sum_{k=1}^{m} a_k$ für $a_k \leqq a$ bedeutet, daß a_k nur dann zum Summenwert zu addieren ist, wenn $a_k \leqq a$ ist. Für $a_k > a$ bleibt der alte Summenwert erhalten.)

3.6. Schreiben Sie ein Programm zur Berechnung von

$p = a_1 a_2^2 a_3 a_4^2 a_5 \ldots a_{n-1} a_n^2$ (n gerade) und

$q = \sqrt[n]{1 + |p|}$.

Die Werte a_k sollen in der Reihenfolge $a_1, a_2, \ldots, a_n$ eingegeben werden. Testen Sie dieses Programm, und berechnen Sie damit p und q für

a_k	1,45	2,38	−0,84	1,12	−3,48	0,76	5,42	−1,92

3.7. Was wird vom Rechner im folgenden Programm durchgeführt?

PSS	Taste
000	*CM$_s$
001	STO
002	0
003	RCL
004	0

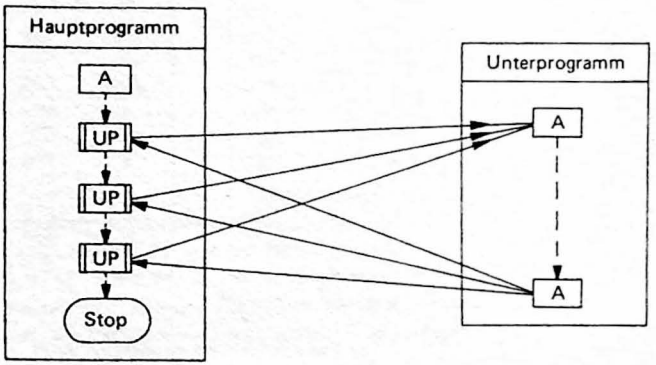
005	×
006	2
007	−
008	1
009	=
010	x^2

011	SUM
012	1
013	*Dsz
014	0
015	0
016	03

017	RCL
018	1
019	R/S
020	RST

4. Unterprogramme

Aufeinanderfolgende Anweisungen treten häufig in derselben Reihenfolge in einem Programm an mehreren Stellen auf. In diesen Fällen erscheint es sinnvoll, diese Anweisungen zusammenzufassen und auf Abruf an der betreffenden Stelle im Programm ausführen zu lassen. Wir nennen eine solche Folge von Anweisungen ein **Unterprogramm (UP)**, das gesamte Programm jetzt auch das Hauptprogramm. Schematisch können wir das eben Beschriebene so darstellen:



Das Symbol [ ] für ein Unterprogramm werden wir auch im Flußdiagramm benutzen

Wie erreichen wir das Verlassen des Hauptprogramms und die Rückkehr? Hier gilt:

> Mit der Anweisung [SBR] n m p oder [SBR] [M] (Subroutine) wird in einem Programm mit den ab der PSS n m p oder den nach [*LBL] [M] stehenden Anweisungen fortgefahren.
> Mit [INV] [SBR] wird die Rückkehr an *die* PSS angeordnet, die auf [SBR] n m p oder [SBR] [M] folgt.

Wichtig: [SBR] n m p benötigt drei; [SBR] [M] zwei und [INV] [SBR] eine PSS.

Beide Anweisungen [SBR] und [INV] [SBR] sind *unbedingte* Sprungbefehle. Man könnte sie ersetzen durch [GTO] n m p, benötigt hierzu allerdings zwei Programmspeicherplätze mehr und verliert eine gewisse Übersicht in der Programmgestaltung.

4.1. Unterprogramme zur Berechnung von Funktionswerten

Wir zeigen die Wirkungsweise von Unterprogrammen am besten am

Beispiel: Es sollen die Funktionswerte

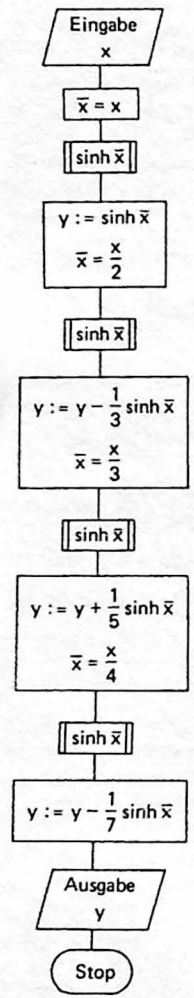
$$y = \sinh x - \frac{1}{3}\sinh\frac{x}{2} + \frac{1}{5}\sinh\frac{x}{3} - \frac{1}{7}\sinh\frac{x}{4}$$

für vorgegebene x-Werte berechnet werden.

Die hyperbolischen Funktionen lassen sich beim TI-58/59 nicht durch einfachen Tastendruck (was nichts anderes als der Aufruf zur Durchführung eines Unterprogramms ist) ermitteln. Wir müssen daher auf die Definitionsgleichung

$\sinh x = \frac{1}{2}(e^x - e^{-x})$ zurückgehen.

Flußdiagramm:



[sinh x̄] bedeutet hier:
Berechne den Wert der hyperbolischen Sinusfunktion für den jeweiligen x̄-Wert, der z.B. im AR oder in einem Speicher R_n zu finden ist. Der x̄-Wert ist bei dieser Aufgabe der Reihe nach:

$x, \frac{x}{2}, \frac{x}{3}, \frac{x}{4}$. Steht der x̄-Wert im AR, so lautet das Programm zur Berechnung von sinh x̄:

INV
ln x
−
1/x
=
÷
2
=

Dieses Unterprogramm wird 4-mal im Hauptprogramm aufgerufen und durchgeführt.

Das gesamte Programm sieht folgendermaßen aus:

PSS	Taste
000	*CM,
001	STO
002	1
003	SBR
004	0
005	53
006	SUM
007	2
008	RCL
009	1
010	÷
011	2
012	=
013	SBR
014	0

015	53
016	÷
017	3
018	+/−
019	=
020	SUM
021	2
022	RCL
023	1
024	÷
025	3
026	=
027	SBR
028	0
029	53
030	÷

031	5
032	=
033	SUM
034	2
035	RCL
036	1
037	÷
038	4
039	=
040	SBR
041	0
042	53
043	÷
044	7
045	+/−
046	=

047	SUM
048	2
049	RCL
050	2
051	R/S
052	RST
053	INV
054	ln x
055	−
056	1/x
057	=
058	÷
059	2
060	=
061	INV SBR

Die auf die Anweisung [SBR] folgende Programmspeicherstelle (hier 053) kann natürlich erst nach Fertigstellung des gesamten Programms angegeben werden.

Benutzeranleitung:

Berechnung von Funktionswerten				
Speicherplan		Eingabe	Taste	Anzeige
1	x	x	R/S	y
2	y	x	R/S	y
		...	usw.	...

Zahlenbeispiel (auf 3 Nachkommastellen):

x	0	0,4	0,8	1,0	1,5	2
y	0	0,356	0,776	1,033	1,905	3,304

Sehr häufig werden Unterprogramme in der folgenden Art benutzt. Für das Verfahren zur allgemeinen Lösung eines Problems (z.B. Berechnung des Maximums einer Funktion $y = f(x)$) existiert ein Programm. Wollen wir dieses in einem besonderen Fall (z.B. für $f(x) = 3 + 2x - x^3$) anwenden, so brauchen wir nur über ein Unterprogramm die Anweisungen zur Berechnung der Funktionswerte hinzuzufügen. Der Arbeitsaufwand kann dadurch erheblich gesenkt werden[1]. Zur Erläuterung geben wir zwei Beispiele, die in ähnlicher Form später bei den Anwendungen wiederholt auftreten werden.

[1] Dieses gilt natürlich ganz besonders für programmierbare Taschenrechner, bei denen statt der manuellen Eingabe ein fertiges Programm mit einer Magnetkarte eingelesen werden kann.

Beispiel: Es soll ein Programm zur Berechnung der Summenwerte

$$s_k = \sum_{j=1}^{k} a_j \quad \text{und} \quad b_n = \frac{1}{n} \sum_{k=1}^{n} s_k = \frac{1}{n} \sum_{k=1}^{n} \sum_{j=1}^{k} a_j$$

geschrieben werden. Angezeigt werden sollen nacheinander s_n und b_n. Die Berechnung der a_k soll über ein Unterprogramm erfolgen.

Nennen wir $\sum_{j=1}^{k} s_j = S_k$, so sieht das Rechenschema folgendermaßen aus:

$s_1 = a_1$; $S_1 = s_1$;

$s_2 = a_1 + a_2 = s_1 + a_2$; $S_2 = s_1 + s_2 = S_1 + s_2$;

$s_3 = a_1 + a_2 + a_3 = s_2 + a_3$; $S_3 = s_1 + s_2 + s_3 = S_2 + s_3$ usw.

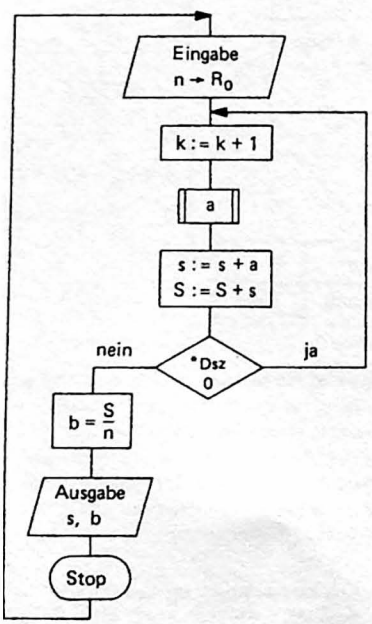
Allgemein erhalten wir die Rekursionsformeln

$s_k = s_{k-1} + a_k$ und $S_k = S_{k-1} + s_k$

oder in der Programmierschreibweise mit dem ‚Ergibt-Zeichen'

$s := s + a$ und $S := S + s$.

Wir entwickeln zunächst das *Flußdiagramm:*



Das Programm lautet (n steht zum Schluß der Rechnung in R_1):

PSS	Taste
000	*CMs
001	STO
002	00
003	1
004	SUM
005	1
006	SBR

007	0
008	28
009	SUM
010	2
011	RCL
012	2
013	SUM
014	3

015	*Dsz
016	0
017	0
018	03
019	R/S
020	RCL
021	3
022	÷

023	RCL
024	1
025	=
026	R/S
027	RST
028	
029	
030	

Benutzeranleitung:

Berechnen von Summenwerten

1. Programm eintasten.
2. BAR auf PSS 028 stellen.
3. Programm zur Berechnung der a_k eintasten. Der Index k steht im Speicher R_1. Tastenfolge mit [INV] [SBR] abschließen und mit [LRN] [RST] in die Betriebsart RECHNEN schalten.

Speicherplan		Eingabe	Taste	Anzeige
0	n, ..., 0	n	R/S	s_n
1	0,...k..., n	–	R/S	b_n
2	s	n	R/S	s_n
3	S	...	usw.	...

Wir erinnern noch einmal an die Einstellung des Befehlsadreßregisters auf die PSS n m p (s. 2.3.): [GTO] n m p [LRN].

Wir zeigen an drei Beispielen für a_k, wie wir das Programm zu benutzen haben. Wir schreiben jeweils die Tastenfolge für die Eingabe nach Punkt 3. der Benutzeranleitung.

a) $a_k = 1$: 1 [INV] [SBR] [LRN] [RST]

b) $a_k = k$: [RCL] 1 [INV] [SBR] [LRN] [RST]

c) $a_k = \frac{1}{k}$: [RCL] 1 [1/x] [INV] [SBR] [LRN] [RST]

Die Durchführung der Rechnung für einige n ergibt die Zahlentabelle

n	$a_k = 1$		$a_k = k$		$a_k = \frac{1}{k}$	
	s_n	b_n	s_n	b_n	s_n	b_n
1	1	1	1	1	1	1
2	2	1,5	3	2	1,5	1,25
5	5	3	15	7	2,283333	1,74
10	10	5,5	55	22	2,928968	2,221865

Im Fall a) können wir die Rechnung leicht ohne Rechner durchführen.

$s_k = \sum_{j=1}^{k} 1 = k$ und $b_n = \frac{1}{n} \sum_{k=1}^{n} k = \frac{1}{n} \cdot \frac{n(n+1)}{2} = \frac{n+1}{2}$. Beim Testen eines aufgestellten Programms ist es sehr wichtig, wenn bekannte Fälle mit dem Programm durchgerechnet werden. Auf diese Art werden häufig Fehler entdeckt.[1)]

Sind die Folgenglieder a_k nicht nach einem analytischen Bildungsgesetz, sondern numerisch gegeben, so wird die Tastenfolge besonders einfach:

[R/S] [INV] [SBR] [LRN] [RST].

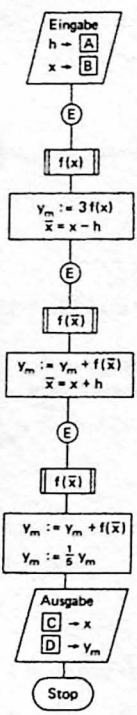
Nach dem Aufruf durch das Unterprogramm stoppt der Rechner, um uns Zeit für die Eingabe des a_k-Wertes zu lassen.

Beispiel: Für eine Funktion $y = f(x)$ soll der Mittelwert

$$y_m = \frac{1}{5}[f(x-h) + 3f(x) + f(x+h)]$$

für gegebene Werte x und h berechnet werden. Das Programm soll mit einem Unterprogramm zur Berechnung der Funktionswerte $y = f(x)$ geschrieben werden.

Flußdiagramm:



Programm
(Speicherplan s. Benutzeranleitung):

PSS	Taste
000	*LBL
001	A
002	STO
003	2
004	R/S
005	*LBL
006	B
007	STO
008	1
009	SBR
010	E
011	x
012	3
013	=
014	SUM
015	3
016	RCL
017	1
018	−
019	RCL
020	2
021	=
022	SBR
023	E
024	SUM
025	3

026	RCL
027	1
028	+
029	RCL
030	2
031	=
032	SBR
033	E
034	SUM
035	03
036	5
037	INV
038	*Prd
039	3
040	R/S
041	*LBL
042	C
043	RCL
044	1
045	R/S
046	*LBL
047	D
048	RCL
049	3
050	R/S
051	*LBL
052	E

1) Für mathematisch Interessierte: Auch im Fall b) können wir die Ergebnisse formelmäßig angeben. Versuchen Sie, die Formeln $s_n = \frac{n(n+1)}{2}$ und $b_n = \frac{(n+1)(n+2)}{6}$ zu bestätigen.

Benutzeranleitung:

Mittelwertberechnung

1. Programm eintasten.
2. Tasten [GTO] [E] [LRN] drücken.
3. Programm zur Berechnung der Funktionswerte $y = f(x)$ eintasten. Der jeweilige x-Wert steht im AR. Tastenfolge mit [INV] [SBR] abschließen und mit [LRN] in die Betriebsart RECHNEN schalten.
4. Vor der Eingabe neuer h- und x-Werte [$^{\bullet}CM_s$] betätigen.

Speicherplan		Eingabe	Taste	Anzeige
1	x	h	A	–
2	h	x	B	–
3	y_m	–	C	x
		–	D	y_m
		h	A	–
		...	usw.	...

Wir testen das Programm mit den beiden Funktionen

a) $f(x) = 1$: 1 [INV] [SBR] [LRN]

Anzeige: x und $y_m = 1$.

b) $f(x) = 2x$: [x] 2 [=] [INV] [SBR] [LRN]

Anzeige: x und $y_m = 2x$.

Für die Funktion $f(x) = \sqrt{x}\ \ln(1 + x^2)$ geben wir als Unterprogramm ein:

[STO] 4 [$\sqrt{x}$] [x] [(] 1 [+] [RCL] 4 [x^2] [)] [ln x] [=]

[INV] [SBR] [LRN]

Hier müssen wir den Wert x = (AR) zunächst speichern ($x \rightarrow R_4$), weil wir ihn später bei x^2 erneut benötigen. Der Wert, für den f(x) berechnet werden soll, steht *nur* im AR, in R_1 steht ein anderer Wert.

Für die obige Funktion erhalten wir z.B. die folgende Zahlentabelle (mit [$^{\bullet}$Fix] 5):

h	0,5	1	0,1
x	1	2	3
y_m	0,73616	2,30192	3,98811

Bemerkungen über Unterprogramme

1) Wann lohnen sich Unterprogramme? Zunächst einmal sicherlich immer dann, wenn Programme von der Art der letzten beiden Beispiele häufiger mit wechselnden Funktionen auftreten. Haben wir dagegen ein Problem nur einmal zu behandeln, so werden sich Unter-

programme als Zusammenfassung von Anweisungsfolgen nur lohnen, wenn diese genügend oft auftreten oder hinreichend lang sind. Treten in einem Programm n Programmschritte in gleicher Reihenfolge k mal auf, so sind dieses k n Programmschritte. Mit einem Unterprogramm [SBR] [M] würden sich insgesamt $2k + n + 3$ Programmschritte ergeben. Die 3 ergibt sich aus [*LBL], [M] und [INV] [SBR]. Wir sparen somit bei der Benutzung von Unterprogrammen Speicherplätze im Falle

$2k + n + 3 < kn$.

In der folgenden Tabelle stellen wir einige Ergebnisse über die Anzahl der Programmschritte mit oder ohne Unterprogramm zusammen.

k	n	ohne UP	mit UP
2	7	14	14
	8	16	15
	9	18	16
	10	20	17
3	5	15	14
	6	18	15
	7	21	16
	8	24	17
4	4	16	15
	5	20	16
	6	24	17
	7	28	18

k	n	ohne UP	mit UP
5	4	20	17
	5	25	18
	6	30	19
6	4	24	19
	5	30	20
7	4	28	21
	5	35	22
8	4	32	23
	5	40	24

2) Darf mit [GTO] oder einer Vergleichsanweisung, z.B. [*x ≧ t], in ein Unterprogramm gesprungen werden? Die Antwort lautet: **Nein!** Der Sprung in ein Unterprogramm wird zwar ausgeführt und auch die folgenden Anweisungen bis [INV] [SBR], dann aber stoppt der Rechner. Es findet keine Rückkehr ins Hauptprogramm statt.

000	GTO	008	=
001	0	009	R/S
002	10	010	1
003	SBR	011	+
004	0	012	2
005	10	013	=
006	+	014	INV SBR
007	4	015	RST

Sehen Sie sich dazu das nebenstehende Programm an. Überlegen Sie, was Ihr Rechner nach Betätigen der Taste [R/S] anzeigen wird. Überprüfen Sie die Richtigkeit Ihrer Voraussage. [INV] [SBR] wird in diesem Fall als eine Stop-Anweisung aufgefaßt. Stellen Sie dagegen das BAR mit [GTO] 003 auf die PSS 003 und betätigen [R/S], dann wird die Anweisung [INV] [SBR] ausgeführt und Sie erhalten in der Anzeige 7.

000	SBR	006	1
001	0	007	+
002	06	008	2
003	+	009	=
004	4	010	INV SBR
005	=	011	RST

Untersuchen Sie das nebenstehende Programm. Der Rechner zeigt hier das Ergebnis 3 an. Es sieht so aus, als hätte er den [INV] [SBR]-Befehl nicht ausgeführt. Wenn Sie aber das Programm mit [SST] in der Betriebsart RECHNEN durchgehen, dann erscheint zunächst 1 + 2 + 4 = 7, danach erst wieder 1 + 2 = 3. Erst beim 2. Erreichen von [INV] [SBR] stoppt der Rechner.

3) Darf mit [GTO] oder einer Vergleichsanweisung aus einem Unterprogramm zurück ins Hauptprogramm gesprungen werden? Die Antwort lautet: **Ja!**

000	SBR	009	+
001	0	010	3
002	08	011	=
003	+	012	$*x \geq t$
004	5	013	0
005	=	014	06
006	R/S	015	$\sqrt{x}$
007	RST	016	INV SBR
008	1		

Im nebenstehenden Programm werden nach Aufruf des Unterprogramms durch [SBR] 008 die Anweisungen ab der PSS 008 ausgeführt, also zunächst 1 + 3 = 4. Diese Zahl wird durch die Anweisung in der PSS 012 mit dem Wert im T-Speicher verglichen. Ist $4 \geq (T)$, so wird die Anweisung in der PSS 006 ausgeführt, der Rechner stoppt und zeigt 4 an. Für $4 < (T)$ wird mit [$\sqrt{x}$] in der PSS 015 fortgefahren. Auch hier stoppt das Programm an der PSS 006, zeigt jetzt aber $\sqrt{4} + 5 = 7$ an.

4) Vorsicht ist in einem Unterprogramm bei der Benutzung der [=]-Anweisung geboten, die in eine unvollständige Operation gesetzt wurde.

000	RCL	009	RCL
001	1	010	2
002	+	011	=
003	SBR	012	÷
004	0	013	RCL
005	07	014	2
006	R/S	015	=
007	1	016	INV SBR
008	+		

Nehmen wir an, $z = y + f(x)$ mit $f(x) = \frac{1+x}{x}$ soll berechnet werden. Schreiben wir das Programm in der nebenstehenden Form und geben $y = 1 \rightarrow R_1$ und $x = 2 \rightarrow R_2$ ein, so zeigt der Rechner den Wert 2 an. Es beträgt aber $z = 1 + \frac{3}{2} = 2{,}5$. Der Fehler ist hier leicht zu erkennen. Nach dem Programm wird gerechnet $1 + 1 \div 2 = 4$ und $\frac{4}{2} = 2$. Das [=]-Zeichen im Unterprogramm schließt die im Hauptprogramm stehende unvollständige Operation ab. Das Programm zur richtigen Berechnung von z muß z.B. 1 [+] [RCL] 2 in Klammern setzen und [=] in der PSS 011 löschen.

4.2. Unterprogramme als Zusammenfassung einer Folge gleicher Anweisungen

Bisher haben wir Unterprogramme verwendet, um mit ihnen Funktionswerte zu berechnen. Sie könnten auch als selbständige Programme benutzt werden. Unterprogramme können aber auch ganz anders zustandekommen. Sehen wir uns dazu noch einmal das Beispiel der Zinsberechnung in 3.3. an. Die PSS 020 bis 026, 031 bis 037, 042 bis 048 und 053 bis 059 bestehen aus einer Folge gleicher Anweisungen. Diese können wir ebenfalls zu einem Unterprogramm zusammenfassen. Die Anweisungen von der PSS 009 bis 015 können wir einbeziehen, wenn wir die 1 im Vorwege in den Datenspeicher R_6 bringen. Berechnen wir die auf dieses Unterprogramm folgenden Zahlen 2, 3, 4, 5 im T-Speicher noch durch die Folge [x ⇄ t] [+] 1 [=] [x ⇄ t], so schließt sich [x ⇄ t] des früheren Programms ebenfalls an die bisherige Anweisungsfolge an.

Das Programm zur Zinsberechnung lautet somit:

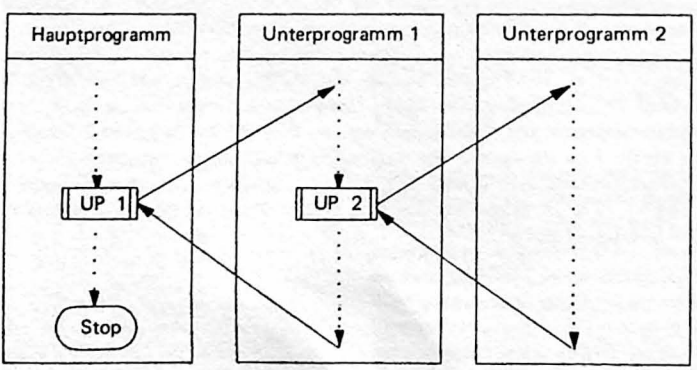
PSS	Taste
000	STO
001	0
002	R/S
003	STO
004	07
005	1
006	STO
007	6
008	x ⇆ t
009	RCL
010	1
011	SBR
012	0
013	60
014	RCL
015	2
016	SBR
017	0
018	60
019	RCL
020	3
021	SBR
022	0
023	60
024	RCL
025	4
026	SBR
027	0
028	60
029	RCL
030	5
031	SBR
032	0
033	60
034	÷
035	0
036	=
037	RCL
038	0
039	×
040	RCL
041	6
042	=
043	R/S
044	RCL
045	6
046	INV
047	y^x
048	RCL
049	7
050	−
051	1
052	=
053	×
054	1
055	0
056	0
057	=
058	R/S
059	RST
060	*Prd
061	6
062	RCL
063	7
064	*x = t
065	0
066	37
067	x ⇆ t
068	÷
069	1
070	=
071	x ⇆ t
072	INV SBR

Der Versuch, auch [RCL] noch in das Unterprogramm zu nehmen, mißlingt leider. In der Anweisung [RCL] n dürfen [RCL] und n **nicht** getrennt werden. Ebenso ist es natürlich bei [STO] n, [SUM] n usw. Mit dem obigen Unterprogramm haben wir das Programm aus 3.3. von 86 Programmschritten auf 73 reduzieren können. Überprüfen Sie selbst, ob Sie mit diesem Programm auf dieselben Werte wie die in der Tabelle in 3.3. kommen.

Ein weiteres Beispiel zu Unterprogrammen dieser Art finden Sie im nächsten Abschnitt.

4.3. Unterprogramme höherer Stufe

Zuweilen treten in einem Unterprogramm Anweisungsfolgen auf, die wir ebenfalls als Unterprogramme auffassen können. Schematisch dargestellt sieht dieses folgendermaßen aus:



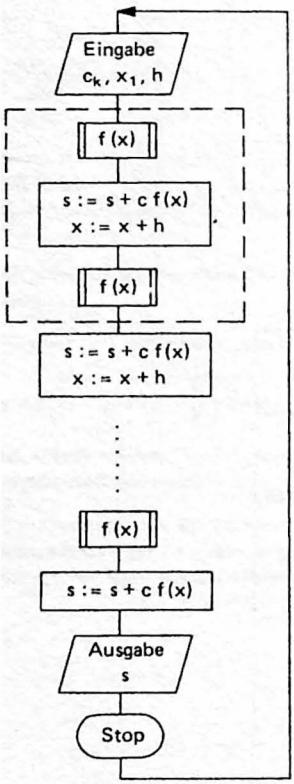
Wir nennen das UP 2 ein Unterprogramm 2. Stufe. Der TI-58/59 kann Unterprogramme bis zur 6. Stufe (was in praktischen Fällen beim programmierbaren Taschenrechner kaum vorkommen dürfte) in richtiger Reihenfolge verarbeiten. Die Programmierung sieht in diesen Fällen genauso aus wie bei den Unterprogrammen 1. Stufe. Jedes Unterprogramm wird durch [SBR] n m p oder [SBR] [M] aufgerufen und mit [INV] [SBR] abgeschlossen.

Beispiel: Es soll ein Programm zur Berechnung der Summe

$$s = \sum_{k=1}^{6} c_k f(x_k) \quad \text{mit} \quad x_k = x_1 + (k-1)\,h \qquad (k \in \mathbb{N}_6)$$

für eine beliebige Funktion f(x) und beliebige Konstanten c_k geschrieben werden.

Für den gestrichelt eingerahmten Ausschnitt des *Flußdiagramms* sind die Anweisungen unten rechts angegeben (Speicherplan s. Benutzeranleitung).




```
SBR
E
x
RCL
1
=
SUM
9
RCL
8
SUM
7
SBR
E
x
```

Die im Rechteck eingeschlossene Anweisungsfolge erscheint im gesamten Programm fünfmal. Wir schreiben sie daher als ein Unterprogramm ([SBR] [D]), das [SBR] [E] zur Berechnung von f(x) als Unterprogramm 2. Stufe enthält. Summieren wir das erste Mal noch 0 in den Speicher R_9 und sorgen dafür, daß vor Beginn der Rechnung $x_1 - h$ in R_7 steht, so können wir auch beim ersten Mal [f(x)] in das Unterprogramm hineinnehmen.

Für die Speicherung der Konstanten c_k mit der Programmadreßtaste [A] benutzen wir die indirekte Adressierung (s. 2.5.). x_1 wird mit [B] und h mit [C] in die festgelegten Speicher gebracht. Damit lautet das Programm zur Summenberechnung:

PSS	Taste
000	*LBL
001	A
002	STO *Ind
003	00
004	1
005	SUM
006	0
007	R/S
008	*LBL
009	B
010	STO
011	7
012	R/S
013	*LBL
014	C
015	STO
016	8

017	INV
018	SUM
019	07
020	0
021	STO
022	9
023	SBR
024	D
025	RCL
026	1
027	SBR
028	D
029	RCL
030	2
031	SBR
032	D
033	RCL
034	3

035	SBR
036	D
037	RCL
038	4
039	SBR
040	D
041	RCL
042	5
043	SBR
044	D
045	RCL
046	6
047	=
048	SUM
049	9
050	RCL
051	9
052	R/S

053	*LBL
054	D
055	=
056	SUM
057	9
058	RCL
059	8
060	SUM
061	7
062	SBR
063	E
064	×
065	INV SBR
066	*LBL
067	E
068	
069	
070	

Benutzeranleitung:

Berechnung der Summe $s = \sum_{k=1}^{6} c_k f(x_k)$.

1. [GTO] [E] [LRN] betätigen.
2. Programm zur Berechnung der Funktionswerte f(x) eintasten. Tastenfolge mit [INV] [SBR] abschließen und mit [LRN] in die Betriebsart RECHNEN schalten.

Speicherplan		Eingabe	Taste	Anzeige
0	0, 1, ..., 6	0	*CM$_s$ A	–
1	c_1	c_1	A	–
2	c_2	c_2	A	–
3	c_3	c_3	A	–
4	c_4	c_4	A	–
5	c_5	c_5	A	–
6	c_6	c_6	A	–
7	x	x_1	B	–
8	h	h	C	s
9	s	x_1	B	–
		h	usw.	–

Wir testen das Programm mit

a) $c_k = k$, $f(x) = 1$, x_1 und h beliebig: $s = \sum_{k=1}^{6} k = 21$;

b) $c_k = \frac{1}{k}$, $f(x) = x^3$, $x_1 = 1$, $h = 1$: $s = \sum_{k=1}^{6} \frac{1}{k} k^3 = 91$.

Nach erfolgreichem Test berechnen wir s für

$$f(x) = \sqrt{1 + \sin^2 \frac{\pi x}{3}} + e^{-x}, \quad c = (c_k) = (1; -2; 1{,}5; 2; -1; 3),$$

$x_1 = 0$ und $h = 0{,}25$.

Tastenfolge für f(x): [RCL] 7 [x] [•π] [÷] 3 [=] [•sin] [x²] [+] 1 [=] [√x]

[+] [RCL] 7 [+/−] [INV] [ln x] [=]

Vergessen Sie nicht die Taste [•Rad]!

Ergebnis: s = 7,697310818.

Bevor wir dieses Beispiel endgültig verlassen, wollen wir gestehen, daß nicht unbedingt ein UP 2. Stufe benutzt werden muß. Etwas einfacher und kürzer wird das Programm mit der Sprunganweisung [•Dsz]. Versuchen Sie es einmal! Sie müssen allerdings die Eingabe der c_k-Werte ändern.

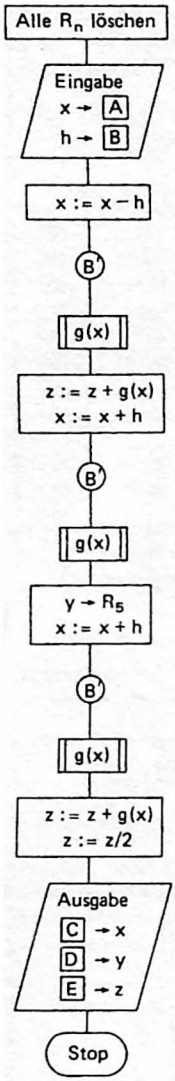
Ein vom Aufbau her sicherlich nicht ganz einfaches Programm mit einem UP 2. Stufe zeigt das

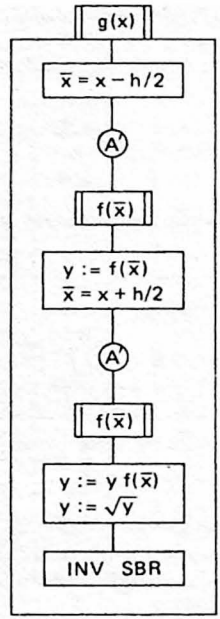
Beispiel: Für gegebene x- und h-Werte sollen

$$y = g(x) = \sqrt{f\left(x - \frac{h}{2}\right) \cdot f\left(x + \frac{h}{2}\right)} \quad \text{und} \quad z = \frac{1}{2}[g(x - h) + g(x + h)]$$

berechnet werden. Das Programm ist für eine beliebige Funktion f(x) zu schreiben. Ausgegeben werden sollen x, y, z.

Das *Flußdiagramm* für das Hauptprogramm und Unterprogramm zur Berechnung des Funktionswertes g(x) ist unten dargestellt. Beachten Sie dabei: In g(x) und f(x) bedeutet x immer *den* Wert, der unmittelbar der Darstellung [g(x)] oder [f(x)] vorangeht.





Mit dem in der Benutzeranleitung angegebenen Speicherplan und der Eingabe- und Ausgabefestlegung im Flußdiagramm schreiben wir das Programm.

PSS	Taste
000	*LBL
001	A
002	*CM$_s$
003	STO
004	1
005	−
006	R/S
007	*LBL
008	B
009	STO
010	2
011	=
012	STO
013	3
014	SBR
015	*B′
016	SUM
017	6
018	RCL
019	2
020	SUM

021	3
022	SBR
023	*B′
024	STO
025	5
026	RCL
027	2
028	SUM
029	3
030	SBR
031	*B′
032	SUM
033	06
034	2
035	INV
036	*Prd
037	6
038	R/S
039	*LBL
040	*B′
041	RCL
042	3

043	−
044	RCL
045	2
046	÷
047	2
048	=
049	SBR
050	*A′
051	STO
052	4
053	RCL
054	3
055	+
056	RCL
057	2
058	÷
059	2
060	=
061	SBR
062	*A′
063	*Prd
064	4

065	RCL
066	4
067	$\sqrt{x}$
068	INV SBR
069	*LBL
070	C
071	RCL
072	1
073	R/S
074	*LBL
075	D
076	RCL
077	5
078	R/S
079	*LBL
080	E
081	RCL
082	6
083	R/S
084	*LBL
085	*A′
086	

Benutzeranleitung:

Berechnung von Funktionswerten

1. GTO *A′ LRN betätigen.
2. Programm zur Berechnung von f(x) eingeben. Der x-Wert steht im AR. Tastenfolge mit INV SBR abschließen und mit LRN in die Betriebsart RECHNEN schalten.

Speicherplan		Eingabe	Taste	Anzeige
1	x	x	A	−
2	h	h	B	−
3	x−h, x, x+h	−	C	x
4	g(x)	−	D	y
5	y	−	E	z
6	z			

Wir testen das Programm mit

a) f(x) = 1, x und h beliebig: y = g(x) = 1 und z = 1;

b) $f(x) = x^2$: $y = g(x) = x^2 - \left(\frac{h}{2}\right)^2$ und $z = x^2 + \frac{3}{4}h^2$. Für z.B. x = 3 und h = 2 werden y = 8 und z = 12.

Nach erfolgreichem Test können wir das Programm für beliebige Funktionen f(x) benutzen, z.B. für

$$f(x) = x + \sqrt{x^2 + 1} + |e^{-3x} - 2|.$$

Entwickeln Sie die Tastenfolge für f(x), geben Sie diese nach der Benutzeranleitung ein, und überprüfen Sie mit dem Programm die folgende Tabelle:

x	h	y	z
1,5	0,2	5,288728953	5,290036148
4	0,5	10,11155743	10,11330424
− 2	1	399,407695	4057,685641
0	0,1	1,979845274	1,936088882

4.4. Übungsaufgaben

4.1. Schreiben Sie ein Programm zur Berechnung des Summenwertes

$$s = \sum_{k=1}^{n} \left(1 + \frac{1}{2} a_k^2\right) \qquad \text{für beliebige } a_k .$$

Testen Sie das Programm mit a) $a_k = 1$ und b) $a_k = k$. Berechnen Sie mit dem Programm s auf 4 Nachkommastellen für $n = 6$ und

$$a_k = \frac{k}{1 + k^2} e^{-\frac{k}{10}} .$$

4.2. Für gegebene Zahlenpaare (x, y) sollen

$$z_1 = 2\sqrt{x^2 + y^2} + \frac{1}{3}\sqrt{(x+1)^2 + y^2},$$

$$z_2 = \sqrt{(x+1)^2 + (y+1)^2} + \ln\left(1 + \sqrt{(x+1)^2 + y^2}\right),$$

$$z_3 = \sqrt{z_1^2 + z_2^2}$$

auf 3 Nachkommastellen berechnet werden. Zahlenbeispiel (die ersten Wertepaare dienen als Test):

x	0	0	− 1	2,74	− 3,65	0,468
y	0	1	1	1,28	2,83	− 0,702

4.3. Schreiben Sie ein Programm zur Berechnung von

$$s = \sum_{k=1}^{n} a_k \left(1 + \frac{1}{n} b_k\right) \quad \text{und} \quad c = \frac{s}{n\,(n+1)}$$

für beliebige a_k und b_k. Reservieren Sie für die Unterprogramme zur Berechnung von a_k und b_k jeweils etwa 100 Programmspeicher.

Testen Sie das Programm mit

a) $a_k = 1$ und $b_k = 1$ $(s = n + 1$ und $c = \frac{1}{n})$ und

b) $a_k = k$ und $b_k = k \left(s = \frac{(n+1)(5n+1)}{6} \text{ und } c = \frac{5n+1}{6n} \right)$.

Berechnen Sie nach erfolgreichem Test s und c für

$$a_k = \frac{1}{1 + e^{k/5}}, \quad b_k = \sin\frac{k\pi}{6} \quad \text{und} \quad n = 5, 10, 20.$$

4.4. Schreiben Sie ein Programm zur Berechnung des Mittelwertes

$$y_m = \frac{1}{9}\left[f(x-h) + 2f\left(x - \frac{h}{2}\right) + 3f(x) + 2f\left(x + \frac{h}{2}\right) + f(x+h)\right]$$

für eine beliebige Funktion f(x).

Testen Sie das Programm mit

a) $f(x) = 1$ und b) $f(x) = 2x$.

Berechnen Sie nach dem Test y_m für

$$f(x) = \frac{1}{1 + x^2} + \arcsin\sqrt{1 - x^2}$$

und die Wertepaare

$(x; h) \in \{(0; 0{,}1),\ (0{,}5; 0{,}5),\ (-0{,}6; 0{,}25)\}$.

4.5. Versuchen Sie herauszufinden, was mit dem folgenden Programm berechnet wird (nennen Sie $(R_1) = x$):

PSS	Taste
000	*CM$_s$
001	STO
002	01
003	0
004	SBR
005	x^2
006	INV
007	ln x
008	SUM
009	02
010	1

011	+/−
012	SBR
013	x^2
014	$\sqrt{x}$
015	SUM
016	02
017	2
018	SBR
019	x^2
020	ln x
021	SUM
022	02

023	1
024	SBR
025	x^2
026	1/x
027	SUM
028	2
029	RCL
030	2
031	R/S
032	RST
033	*LBL
034	x^2

035	+
036	RCL
037	1
038	=
039	x^2
040	+
041	1
042	=
043	INV SBR

5. Der Drucker PC-100 A

Eine außerordentlich hilfreiche Ergänzung zum TI-58/59 ist der Drucker PC-100 A. So wie der programmierbare Rechner uns von der langwierigen und oft monotonen Rechenarbeit befreit, so kann uns der Drucker z.B. das Abschreiben von Rechenergebnissen und das erneute manuelle Starten des Rechenablaufs ersparen. Die wenigsten Leser dieses Buches und Besitzer eines TI-58 oder TI-59 werden wohl einen eigenen PC-100 A jederzeit zu Hause zur Verfügung haben doch werden wohl schon jetzt oder in einiger Zeit in einigen Fachbereichen, Instituten, Gymnasien usw. Drucker bereitstehen, an die Sie Ihren programmierbaren Rechner anschließen können. Wir wollen daher in diesem Abschnitt kurz auf einige Eigenschaften des Druckers eingehen. Bei den folgenden Erklärungen gehen wir davon aus, daß Rechner und Drucker zu einer Einheit zusammengeschlossen sind (Bild 5.1).

Bild 5.1

5.1. Auflisten eines Programms und der Datenspeicher

Haben Sie ein Programm in Ihren Rechner eingetastet, so können Sie sich dieses Programm teilweise oder vollständig vom Drucker ausgeben lassen.

> In der Betriebsart RECHNEN wird mit der Taste [*List] das im Rechner gespeicherte Programm von der eingestellten PSS ab in der Form
>
> nnn nn XXX
>
> Anweisung nach Anweisung bis zur letzten PSS ausgegeben. Mit [R/S] kann diese Auflistung jederzeit unterbrochen werden.

Die dreiziffrige Zahl nnn gibt die PSS und die zweiziffrige nn den Tastenkode an. XXX bedeutet das auf der Rechnertaste angegebene Symbol. Hierbei sind folgende Abweichungen zwischen Druckersymbol und Rechnertaste zu beachten:

Rechnertaste	*x = t	*x ≧ t	*If flg	*St flg	*Exc *Ind	GTO *Ind	*Prd *Ind
Druckersymbol	EQ	GE	IFF	STF	EX*	GO*	PD*

RCL *Ind	SUM *Ind	STO *Ind	INV SBR	x ⇆ t	*List	*Write	*Pause
RC*	SM*	ST*	RTN	X ≧ T	LST	WRT	PAU

Alle Tasten mit einem vorangestellten [INV] erhalten im Druckersymbol ein I, z.B. [INV] [*tan] : ITAN. Allerdings werden diese Symbole nicht beim Auflisten eines Programms ausgedruckt, sondern nur bei TRACE (s. weiter unten).

Die Taste [*List] hat nur in der Betriebsart RECHNEN eine Bedeutung. In der Betriebsart RUN wird [*List] im Programm übergangen (hat dort also eine ähnliche Wirkung wie [*Nop]).

Tabelle 5.1.1 zeigt die Auflistung des Programms zur iterativen Quadratwurzelberechnung (s. 2.4.). Mit dieser Liste kann

1. in aller Ruhe die Suche nach einem eventuellen Fehler im Programm durchgeführt und
2. auf bequeme Weise eine Kopie des Programms hergestellt werden.

Eine Aufzeichnung des Programms in nicht-kodierter Form *und* der Zahlenwerte des Anzeigeregisters kann mit der Drucker-Taste **TRACE** erreicht werden. Diese Taste rastet beim Betätigen ein und gibt solange Ergebnisse und Anweisungen aus, bis sie wieder gelöst wird. Die Taste TRACE kann sowohl beim Programmablauf als auch beim manuellen Rechnen benutzt werden. Tabelle 5.1.2 zeigt die Aufzeichnung des 1. Iterationsschrittes des Programms zur Quadratwurzelberechnung.

Für das Auflisten der Datenspeicher gilt:

Mit [INV] [*List] werden die Inhalte der Datenspeicher von der Adresse ab ausgedruckt, die im AR angegeben ist. Die Auflistung der Daten kann manuell durch [R/S] gestoppt werden.

[INV] [*List] kann sowohl manuell als auch als Anweisung in einem Programm benutzt werden. Für das Beispiel „Lineares Gleichungssystem" in 2.5. erhalten wir mit 0 [INV] [*List] die Tabelle 5.1.3.

Tabelle 5.1.1

000	76	LBL
001	11	A
002	42	STO
003	01	01
004	91	R/S
005	76	LBL
006	12	B
007	42	STO
008	02	02
009	85	+
010	43	RCL
011	01	01
012	55	÷
013	43	RCL
014	02	02
015	95	=
016	55	÷
017	02	2
018	95	=
019	91	R/S
020	00	0

Tabelle 5.1.2

	3.	STO
		2
	3.	
	3.	+
	3.	RCL
		1
14.	256	
14.	256	÷
14.	256	RCL
	3.	
	3.	=
7.	752	
7.	752	÷
	2.	=
3.	876	
		R/S

Tabelle 5.1.3

10.	00
3.74	01
5.61	02
16.72	03
1.89	04
4.53	05
12.34	06
0.	07
6.339	08
1.028	09
2.295	10
0.	11
0.	12

5.2. Das Ausdrucken von Daten

[*Prt] (Print) veranlaßt das Ausdrucken des Zahlenwertes im AR und anschließenden Zeilentransport des Papiers.

[*Adv] (Advance) bewirkt einen Zeilenvorschub des Papiers, also die Herstellung einer Leerzeile.

[*Prt] und [*Adv] können sowohl als Anweisung in einem Programm als auch manuell in der Betriebsart RECHNEN benutzt werden. Falls der Rechner nicht an einen Drucker angeschlossen wird, werden diese Anweisungen wie [*Nop] aufgefaßt.

Programme aus den vorhergehenden Abschnitten können mit der Print-Anweisung sehr leicht so umgeschrieben werden, daß sie jetzt auch in Verbindung mit dem Drucker zu benutzen sind. Sehen wir uns dazu wieder das Beispiel der Quadratwurzelberechnung (s. 2.4.) an. Wir wollen nach jedem Iterationsschritt den x-Wert ausdrucken lassen. Das Programm hierzu ist unten links dargestellt. Wir haben in dem früheren Programm lediglich in der PSS 019 die Anweisung [R/S] durch [*Prt] ersetzt und mit B in der PSS 020 die Rechnung erneut gestartet. Der Rechner durchläuft unentwegt die Schleife im Programm, wir müssen ihn manuell mit [R/S] stoppen. Die ausgedruckten Zahlenwerte sind unten rechts abgebildet.

PSS	Taste
000	*LBL
001	A
002	STO
003	1
004	R/S
005	*LBL
006	B
007	STO
008	2
009	+
010	RCL
011	1
012	÷
013	RCL
014	2
015	=
016	÷
017	2
018	=
019	*Prt
020	B

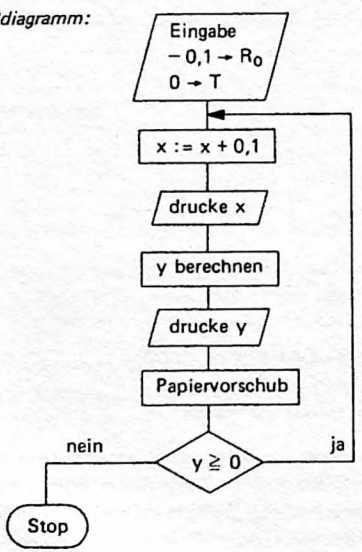
```
         3.876
3.777009288
3.775712076
3.775711853
3.775711853
3.775711853
3.775711853
```

Beispiel: Für x = 0; 0,1; 0,2; ... sollen die Funktionswerte

$$y = 3 \cdot e^{-\frac{x}{2}} - 2 \cdot e^{-\frac{x}{5}}$$

berechnet und die Zahlenpaare (x, y) mit einer Leerzeile zum nächsten Zahlenpaar bis zum 1. negativen y-Wert ausgedruckt werden.

Flußdiagramm:



Das Programm für diese Aufgabe lautet:

PSS	Taste
000	.
001	1
002	+/−
003	STO
004	0
005	*CP
006	.
007	1
008	SUM
009	0

010	RCL
011	0
012	*Prt
013	÷
014	2
015	=
016	+/−
017	INV
018	ln x
019	x
020	3

021	−
022	2
023	x
024	(
025	RCL
026	0
027	÷
028	5
029	)
030	+/−
031	INV

032	ln x
033	=
034	*Prt
035	*Adv
036	*x ≧ t
037	0
038	06
039	R/S

Die folgenden Tabellen zeigen die Auflistung des Programms (Tabelle 5.2.1), die Aufzeichnung der Zahlenwerte des AR und der Anweisungen des Programms für zwei Durchläufe (Tabelle 5.2.2) und die ausgedruckten Ergebnisse der Zahlenpaare (x, y) (Tabelle 5.2.3).

Tabelle 5.2.1

```
000  93   .        021  75   -
001  01   1        022  02   2
002  94  +/-       023  65   ×
003  42  STO       024  53   (
004  00   00       025  43  RCL
005  29  CP        026  00   00
006  93   .        027  55   ÷
007  01   1        028  05   5
008  44  SUM       029  54   )
009  00   00       030  94  +/-
010  43  RCL       031  22  INV
011  00   00       032  23  LNX
012  99  PRT       033  95   =
013  55   ÷        034  99  PRT
014  02   2        035  98  ADV
015  95   =        036  77   GE
016  94  +/-       037  00   00
017  22  INV       038  06   06
018  23  LNX       039  91  R/S
019  65   ×        040  00   0
020  03   3
```

Tabelle 5.2.2

-0.1	STO
	0
-0.1	
	CP
0.1	SUM
	0
0.1	
0.1	RCL
	0
0.	
0.	PRT
0.	÷
2.	=
0.	
0.	ILNX
1.	
1.	×
3.	-
2.	×
2.	(
2.	RCL
	0
0.	
0.	÷
5.	)
0.	
0.	ILNX
1.	
1.	=
1.	
1.	PRT
1.	GE
	6
0.1	SUM
	0
0.1	
0.1	RCL
	0
0.1	
0.1	PRT
0.1	÷
2.	=
0.05	
-0.05	ILNX
.9512294245	
.9512294245	×
3.	-
2.	×
2.	(
2.	RCL
	0
0.1	
0.1	÷
5.	)
0.02	
-0.02	ILNX
.9801986733	
.9801986733	=
.8932909269	
.8932909269	PRT

Tabelle 5.2.3

0.	
1.	
	0.1
	.8932909269
	0.2
	.7929333758
	0.3
	.6985948621
	0.4
	.6099595665
	0.5
	.5267275131
	0.6
	.4486137886
	0.7
	.3753477984
	0.8
	.3066725602
	0.9
	0.242344032
	1.
	0.182130473
	1.1
	.1258118352
	1.2
	.0731791862
	1.3
	.0240341587
	1.4
	-.0218115715

II. Teil:
Programmbeispiele aus der Mathematik und Technik

In diesem II. Teil des Buches werden aus verschiedenen Gebieten einige Aufgaben zusammengestellt, die sich mit Hilfe eines programmierbaren Taschenrechners besonders gut lösen lassen. Beim Entwickeln eines Programms setzen wir die Kenntnis der im I. Teil dargestellten Programmiertechnik voraus. Die Abschnitte 6. und 7. wollen *keine* Programmsammlung irgendeines der behandelten Gebiete sein. Der Leser soll vielmehr anhand der aufgeführten Beispiele lernen, wie ein Problem mathematisch formuliert und in ein Programm umgesetzt werden kann. Mit einem formalen Eintasten eines fertigen Programms sollte sich *der* Leser, der das *Programmieren* lernen möchte, nicht zufrieden geben.

6. Beispiele aus der Mathematik

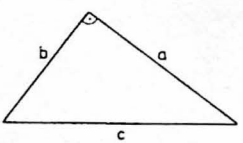
6.1. Zahlen

6.1.1. Pythagoräische Zahlentripel

(a, b, c) wird ein Pythagoräisches Zahlentripel genannt, wenn gilt:

$a^2 + b^2 = c^2$ und $a, b, c \in \mathbb{N}$.

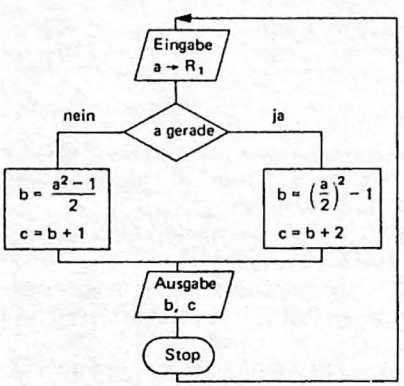
a und b können als Längen der Katheten und c als Länge der Hypotenuse in einem rechtwinkligen Dreieck angesehen werden. Bekannte Pythagoräische Zahlentripel sind z.B. (3, 4, 5) oder (5, 12, 13).



Für $a \in \mathbb{N}$ und $a > 2$ können b und c folgendermaßen berechnet werden[1]:

$$b = \begin{cases} \left(\frac{a}{2}\right)^2 - 1 \\ \frac{a^2-1}{2} \end{cases} \quad \text{und} \quad c = \begin{cases} \left(\frac{a}{2}\right)^2 + 1 = b + 2 & \text{für a gerade;} \\ \frac{a^2+1}{2} = b + 1 & \text{für a ungerade.} \end{cases}$$

Das *Flußdiagramm* für unser Problem sieht sehr einfach aus:



[1] Wir erhalten hiermit nicht *alle* Pythagoräischen Zahlentripel. Diese werden folgendermaßen berechnet: $a = n^2 - m^2$, $b = 2nm$, $c = n^2 + m^2$ mit $n, m \in \mathbb{N}$, $n > m$ und n, m nicht beide gerade und n, m nicht beide ungerade.

Für die Abfrage ‚a gerade' können wir die Funktionstaste [*Int] (Integer) benutzen. Es bedeutet:

Int(x) = größte ganze Zahl, deren Betrag kleiner oder gleich |x| ist.
Oder: Int(x) löscht die Nachkommaziffern der Zahl im AR.

Zum Beispiel Int(4,74) = 4, Int(5) = 5, Int(− 2,32) = − 2.

Mit [INV] [*Int] werden die Nachkommastellen (einschließlich Vorzeichen) der Zahl im AR angezeigt, der ganzzahlige Teil der Zahl wird gelöscht.

Zum Beispiel INV Int(15,784) = 0,784, INV Int($\frac{15}{7}$) = 0,1428571429, INV Int(− 3,048) = − 0,048.

Zurück zu der gestellten Aufgabe. ‚a gerade' bedeutet, daß a ohne Rest durch 2 teilbar ist. Ist a ungerade, so bleibt bei Division durch 2 ein Rest. ‚a gerade' können wir daher auch so formulieren: $\frac{a}{2} - \text{int}(\frac{a}{2}) = 0$. Damit lautet unser Programm:

PSS	Taste
000	R/S
001	STO
002	1
003	÷
004	2
005	−
006	*Int
007	=
008	*x = t

009	0
010	25
011	RCL
012	1
013	x^2
014	−
015	1
016	=
017	÷
018	2

019	=
020	R/S
021	+
022	1
023	=
024	RST
025	RCL
026	1
027	÷
028	2

029	=
030	x^2
031	−
032	1
033	=
034	R/S
035	+
036	2
037	=
038	RST

Eingabe: a [R/S];
Ausgabe: b [R/S] c.

Führen wir mit diesem Programm die Rechnung durch für a = 3, 4, 5, ..., 12, so erhalten wir die folgenden Pythagoräischen Zahlentripel (die nicht teilerfremden, wie z.B. (6, 8, 10), haben wir dabei ausgesondert und nicht mit angegeben):

a	3	5	7	8	9	11	12
b	4	12	24	15	40	60	35
c	5	13	25	17	41	61	37

6.1.2. Größter gemeinsamer Teiler

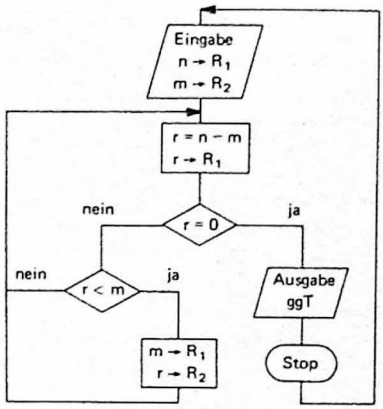
Als größten gemeinsamen Teiler der beiden natürlichen Zahlen n und m (kurz: ggT(n, m)) bezeichnet man die größte Zahl, die sowohl Teiler von n als auch Teiler von m ist. Wir wollen ein Programm zur Berechnung des ggT(n, m) schreiben.

Für $n > m$ gilt offensichtlich ggT(n, m) = ggT(n − m, m). Hiermit läßt sich der größte gemeinsame Teiler nach dem *Euklidischen Algorithmus* so ermitteln:

m wird sooft von n subtrahiert, bis ein Rest $r_1 < m$ bleibt. Ist $r_1 = 0$, so ist ggT(n, m) = m. Für $r_1 \neq 0$ wird r_1 sooft von m subtrahiert, bis ein Rest $r_2 < r_1$ bleibt. Ist $r_2 = 0$, so ist ggT(n, m) = r_1 usw.

Zur Erläuterung diene das folgende Beispiel für n = 84 und m = 35: 84 − 35 = 49 > 35, 49 − 35 = 14 < 35, 35 − 14 = 21 > 14, 21 − 14 = 7 < 14, 14 − 7 = 7 < 14, 7 − 7 = 0, also ggT(84, 35) = 7.

Wir stellen zunächst ein *Flußdiagramm* auf:



Für $m \to R_1$ und $r \to R_2$ hätten wir auch schreiben können: n := m und m := r. Wir haben es hier mit einem Vertauschungsproblem zu tun. Im obigen Zahlenbeispiel hätten wir nach der 2. Subtraktion auch n := 35 und m := 14 setzen und für diese Zahlen nach dem Euklidischen Algorithmus den ggT ermitteln können.

Das Programm zur Berechnung des ggT (n, m) lautet:

PSS	Taste
000	STO
001	1
002	R/S
003	STO
004	02
005	0
006	x ⇆ t
007	RCL
008	1
009	−

010	RCL
011	2
012	=
013	STO
014	1
015	*x = t
016	0
017	38
018	x ⇆ t
019	RCL
020	2

021	x ⇆ t
022	INV
023	*x ≧ t
024	0
025	29
026	GTO
027	0
028	05
029	RCL
030	1
031	*Exc

032	2
033	STO
034	1
035	GTO
036	0
037	05
038	RCL
039	2
040	R/S
041	RST

Eingabe: n [R/S] m; Große Zahl zuerst eingeben?

Ausgabe: ggT (n, m).

Zahlenbeispiele:

n	m	ggT
84	35	7
20	12	4
374	231	11
452	184	4
1023	581	1
7803	2046	3
95139	54033	93

← ca. 25 sec?

6.1.3. Binomialkoeffizienten

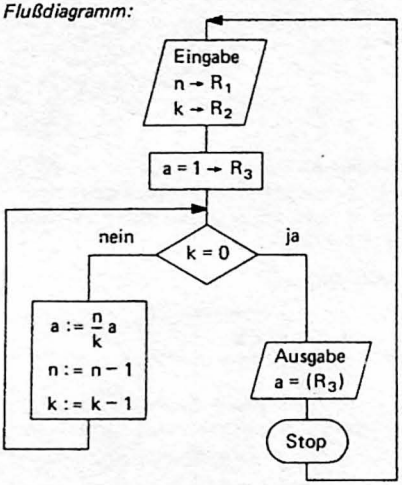
Die Binomialkoeffizienten sind definiert durch

$$a_{n,k} = \binom{n}{k} = \begin{cases} \dfrac{n(n-1)\cdot\ldots\cdot(n-k+1)}{1\cdot 2\cdot 3\cdot\ldots\cdot k} & \text{für } k \in \mathbb{N}; \\ 1 & \text{für } k = 0. \end{cases}$$

Zur Berechnung der $a_{n,k}$ benutzen wir die Rekursionsformel

$$a_{n,k} = \frac{n}{k} a_{n-1,k-1}.$$

Flußdiagramm:



Programm:

PSS	Taste
000	*CP
001	STO
002	1
003	R/S
004	STO
005	02
006	1
007	STO
008	3
009	RCL
010	2
011	*x = t
012	0
013	30
014	1/x
015	x
016	RCL
017	1
018	=
019	*Prd
020	03
021	1
022	+/−
023	SUM
024	1
025	SUM
026	2
027	GTO
028	0
029	09
030	RCL
031	3
032	R/S
033	RST

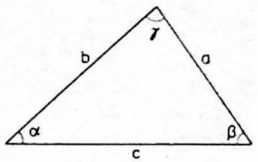
Eingabe: n [R/S] k;

Ausgabe: $\binom{n}{k}$.

Berechnen Sie mit diesem Programm einige Binomialkoeffizienten. Zum Beispiel die Anzahl der Möglichkeiten beim Zahlenlotto (6 Gewinnzahlen werden aus 49 Zahlen ausgelost): $\binom{49}{6} = 13983816$.

6.2. Berechnung des ebenen Dreiecks

Sind von den 6 Größen (3 Seiten und 3 Winkel) eines ebenen Dreiecks 3 (darunter mindestens eine Seite) gegeben, so können die übrigen Größen berechnet werden. Hierzu dienen die Beziehungen:



(1) $\alpha + \beta + \gamma = 180°$;

(2) $\frac{a}{\sin\alpha} = \frac{b}{\sin\beta} = \frac{c}{\sin\gamma}$ (Sinussatz);

(3) $c^2 = a^2 + b^2 - 2ab\cos\gamma$ (Cosinussatz).

Wir stellen in der folgenden Tabelle die möglichen Fälle zusammen. Dabei bedeuten: * gegebene Größe und ° gesuchte Größe.

Fall	a	b	c	α	β	γ	anzuwendender Satz
1) SSS	*	*	*	°	°	°	Cosinussatz
2) SWS	*	*	°	°	°	*	
3) SSW	*	*	°	*	°	°	Sinussatz
4) SWW	*	°	°	*	*	°	
5) WSW	*	°	°	°	*	*	

Den letzten Fall können wir mit $\alpha = 180° - (\beta + \gamma)$ auf den Fall 4) zurückführen. Wir werden ein Programm für die Fälle 1) und 2) (Cosinussatz) und ein 2. Programm für die Fälle 3) und 4) (Sinussatz) entwickeln.

Bei der Anwendung des Sinussatzes müssen wir beachten, daß der Rechner für den Winkel nur den Hauptwert berechnet. Für einen Winkel größer als 90° wird also der falsche Wert verarbeitet. Beachten wir aber, daß z.B. für $a \leqq b$ auch $\alpha \leqq \beta$ ist, so wird die Berechnung von $\alpha < 90°$ auf den richtigen Wert führen.

6.2.1. Cosinussatz

In einem *gemeinsamen* Programm für die Fälle 1) und 2) soll der Rechner den richtigen Lösungsweg selbst herausfinden. Wir können die Unterscheidung der beiden Fälle nach c vornehmen:

> Der Rechner findet im Speicher für die Zahl c den Wert $c \neq 0$ im Fall 1) oder $c = 0$ im Fall 2).

In die nach dem Speicherplan (s. Benutzeranleitung) festgelegten Speicher werden die gesuchten Größen mit Null eingegeben. Im Fall 1) beträgt also $(R_3) \neq 0$ und $(R_6) = 0$, im Fall 2) $(R_3) = 0$ und $(R_6) \neq 0$.

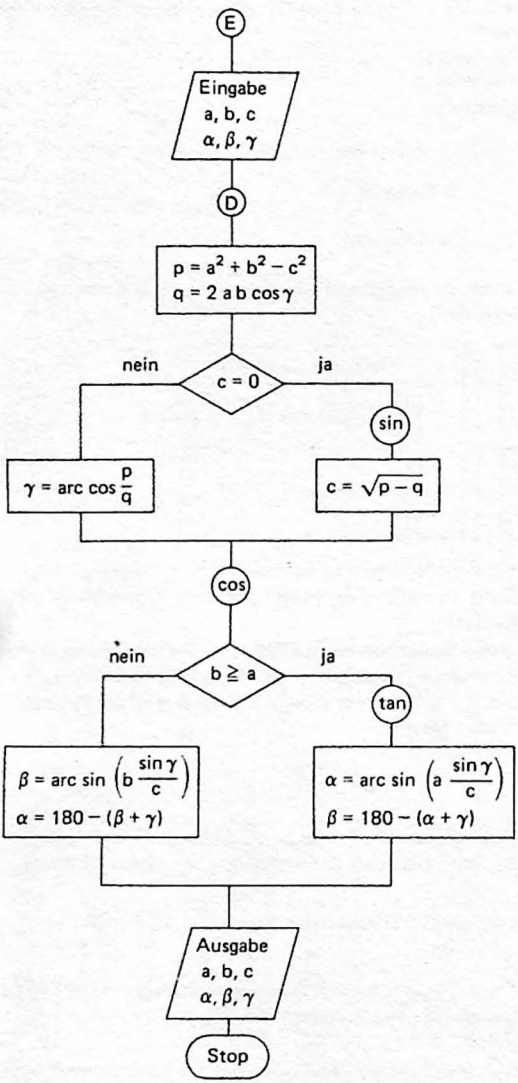
In beiden Fällen treten bei der Anwendung des Cosinussatzes gemeinsame Berechnungen auf, die nur einmal durchgeführt zu werden brauchen. Mit

$p = a^2 + b^2 - c^2$ und $q = 2ab\cos\gamma$

und unter Beachtung, daß im Fall 1) $\gamma = 0°$ ($\cos\gamma = 1$) und im Fall 2) $c = 0$ ist, wird

im Fall 1) $\cos\gamma = \frac{p}{q}$ und im Fall 2) $c = \sqrt{p - q}$.

Nach diesen Überlegungen zeichnen wir das *Flußdiagramm:*



Speicherplan	
0	0, 1, ..., 6
1	a
2	b
3	c
4	α
5	β
6	γ
7	p
8	q
T	0, a

Die Speicherung der eingetasteten Werte a, b, c, α, β, γ nehmen wir indirekt (s. 2.5.) mit [E] vor. Für nicht-gegebene (gesuchte) Größen geben wir 0 ein. Die Ausgabe werden wir einzeln mit Programmadreßtasten vornehmen.

Das Programm lautet:

PSS	Taste
000	*LBL
001	E
002	STO *Ind
003	00
004	1
005	SUM
006	0
007	R/S
008	*LBL
009	D
010	RCL
011	1
012	x^2
013	+
014	RCL
015	2
016	x^2
017	−
018	RCL
019	3
020	x^2
021	=
022	STO
023	07
024	2
025	×
026	RCL
027	1
028	×
029	RCL
030	2
031	×
032	RCL
033	6
034	*cos
035	=
036	STO
037	8
038	*CP
039	RCL
040	3
041	*x = t
042	*sin
043	RCL
044	7
045	÷
046	RCL
047	8
048	=
049	INV
050	*cos
051	STO
052	6
053	GTO
054	*cos
055	*LBL
056	*sin
057	RCL
058	7
059	−
060	RCL
061	8
062	=
063	$\sqrt{x}$
064	STO
065	3
066	*LBL
067	*cos
068	RCL
069	1
070	x ⇆ t
071	RCL
072	2
073	*x ≧ t
074	*tan
075	×
076	RCL
077	6
078	*sin
079	÷
080	RCL
081	3
082	=
083	INV
084	*sin
085	STO
086	5
087	+
088	RCL
089	6
090	−
091	1
092	8
093	0
094	=
095	+/−
096	STO
097	4
098	R/S
099	*LBL
100	*tan
101	x ⇆ t
102	×
103	RCL
104	6
105	*sin
106	÷
107	RCL
108	3
109	=
110	INV
111	*sin
112	STO
113	4
114	+
115	RCL
116	6
117	−
118	1
119	8
120	0
121	=
122	+/−
123	STO
124	5
125	R/S
126	*LBL
127	A
128	RCL
129	1
130	R/S
131	*LBL
132	B
133	RCL
134	2
135	R/S
136	*LBL
137	C
138	RCL
139	3
140	R/S
141	*LBL
142	*A'
143	RCL
144	4
145	R/S
146	*LBL
147	*B'
148	RCL
149	5
150	R/S
151	*LBL
152	*C'
153	RCL
154	6
155	R/S
156	
157	
158	

Eingabe: [*CM$_s$] 0 [E] a [E] b [E] c [E] α [E] β [E] γ [E] [D].
Für nicht-gegebene Größen wird 0 eingetastet.

Ausgabe: [A] a [B] b [C] c [*A'] α [*B'] β [*C'] γ.

Im Anschluß an das Programm für den Sinussatz finden Sie Zahlenbeispiele für das obige Programm.

6.2.2. Sinussatz

In den Fällen 3) und 4) gehen wir bei der Berechnung des Dreiecks vom Sinussatz (2) aus. Für *beide* Fälle wollen wir ein gemeinsames Programm schreiben. Der Rechner soll den Lösungsweg, ähnlich wie vorher beim Cosinussatz, selbst finden. Wir nehmen die Unterscheidung der beiden Fälle nach

$b \neq 0$ (gegeben) oder $b = 0$ (gesucht) vor.

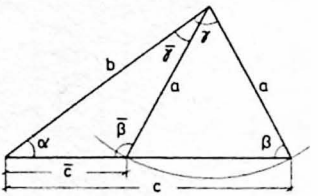


Bild 6.2.1

Den Sinussatz lösen wir entsprechend nach

$$\beta = \arcsin\left(b\,\frac{\sin\alpha}{a}\right) \quad \text{oder}$$

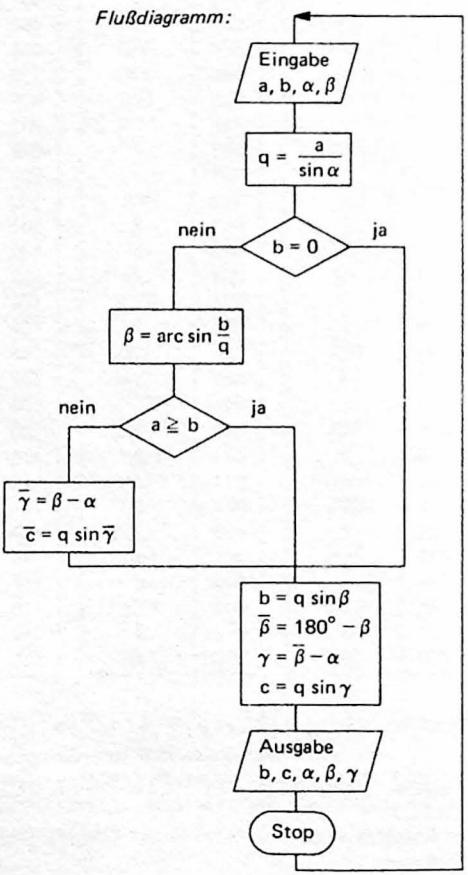
$$b = \frac{a}{\sin\alpha}\sin\beta \quad \text{auf.}$$

Im Fall 3) gibt es für $a < b$ *zwei* Lösungen (s. Bild 6.2.1) oder *keine* Lösung (wird vom Rechner durch Blinken angezeigt). Die Größen der 2. Lösung sollen ebenfalls berechnet werden: $\bar{\beta} = 180° - \beta$, $\bar{\gamma} = \beta - \alpha$,

$$\bar{c} = \frac{a}{\sin\alpha}\sin\bar{\gamma}.$$

Daß die Berechnung von b auch für $b \neq 0$ (also Seite b ist gegeben) und von $\bar{\beta}$ auch für $a < b$ durchgeführt wird, liegt an der Ausnutzung der Programmspeicherplätze. Im ersten Fall hätten wir sonst eine Sprunganweisung GTO hineinnehmen und im zweiten Fall die Berechnung von $180° - \beta$ zweimal durchführen müssen.

Flußdiagramm:



Im Gegensatz zum Programm 6.2.1. wollen wir dieses *ohne* Labels schreiben. Das Programm (Speicherplan s. Benutzeranleitung) zur Dreiecksberechnung lautet somit:

PSS	Taste
000	*CM$_s$
001	*CP
002	STO
003	1
004	R/S
005	STO
006	2
007	R/S
008	STO
009	4
010	R/S
011	STO
012	5
013	RCL
014	1
015	÷
016	RCL
017	4
018	*sin
019	=
020	STO
021	0
022	RCL
023	2
024	*x = t
025	0

026	58
027	÷
028	RCL
029	0
030	=
031	INV
032	*sin
033	STO
034	5
035	RCL
036	2
037	x ◀ t
038	RCL
039	1
040	*x ≧ t
041	0
042	58
043	RCL
044	5
045	−
046	RCL
047	4
048	=
049	STO
050	9
051	*sin
052	x

053	RCL
054	0
055	=
056	STO
057	7
058	RCL
059	5
060	*sin
061	x
062	RCL
063	0
064	=
065	R/S
066	1
067	8
068	0
069	−
070	RCL
071	5
072	=
073	STO
074	8
075	−
076	RCL
077	4
078	=
079	STO

080	6
081	*sin
082	x
083	RCL
084	0
085	=
086	R/S
087	RCL
088	4
089	R/S
090	RCL
091	5
092	R/S
093	RCL
094	6
095	R/S
096	RCL
097	7
098	R/S
099	RCL
100	8
101	R/S
102	RCL
103	9
104	R/S

Dreiecksberechnung: Sinussatz

1. Vor der Eingabe neuer Zahlenwerte das BAR mit [RST] auf 000 stellen und mit [*CM$_s$] alle R$_n$ löschen.
2. Bezeichnungen so vornehmen, daß gegebene Größen mit a, b, α, β benannt werden.
3. Nicht-gegebene Größen (b oder β) sind mit 0 einzutasten.
4. Existiert nur eine Lösung, so wird dieses durch $\bar{c} = 0$ angezeigt.
5. Blinken: Lösung der Aufgabe existiert nicht.

Speicherplan		Eingabe	Taste	Anzeige
0	q	a	R/S	−
1	a	b	R/S	−
2	b	α	R/S	−
3	−	β	R/S	b
4	α	−	R/S	c
5	β	−	R/S	α
6	γ	−	R/S	β
7	$\bar{c}$	−	R/S	γ
8	$\bar{\beta}$	−	R/S	$\bar{c}$
9	$\bar{\gamma}$		R/S	$\bar{\beta}$
			R/S	$\bar{\gamma}$

Zahlenbeispiel (die kursiv gedruckten Werte wurden auf 2 Stellen nach dem Komma berechnet):

a	b	c	α	β	γ
3	4	5	*36,87°*	*53,13°*	*90,00°*
21,6	12,3	16,4	*96,59°*	*34,45°*	*48,96°*
3	4	*5,00*	*36,87°*	*53,13°*	90°
4,28	6,38	*4,99*	*42,04°*	*86,66°*	51,3°
1,28	1,56	2,47	*26,51°*	*32,96°*	*120,53°*
0,84	0,61	*1,25*	*36,20°*	*25,40°*	118,4°
3	4	*5,00*	36,87°	*53,13°*	*90,00°*
		1,40		*126,87°*	*16,26°*
6,82	5,13	*6,31*	72,4°	*45,81°*	*61,79°*
8,06	*11,73*	*12,64*	38,4°	64,7°	*76,90°*
2,37	6,46		46,8°	*keine Lösung*	
34,9	43,8	*66,70*	28,2°	*36,37°*	*115,43°*
		10,50		*143,63°*	*8,17°*
0,46	*0,24*	*0,35*	*99,90°*	31,5°	48,6°

6.3. Berechnung von Funktionswerten

6.3.1. Horner-Schema

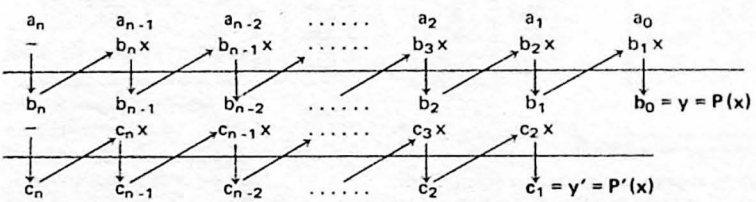
Für ein Polynom n-ten Grades

$y = P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$

lassen sich die Werte der Funktion und deren Ableitung

$y' = P'(x) = n\, a_n x^n + (n-1)\, a_{n-1} x^{n-1} + \ldots + a_1$

nach dem Horner-Schema berechnen[1]:



Wir erhalten z.B. $b_{n-2} = a_{n-2} + b_{n-1} x$ oder $c_{n-2} = b_{n-2} + c_{n-1} x$. Allgemein gilt die Iterationsvorschrift:

$b_k = a_k + b_{k+1} x \;\wedge\; b_{n+1} = 0 \qquad (k \in \mathbb{N}_{0,n})$;

$c_k = b_k + c_{k+1} x \;\wedge\; c_{n+1} = 0 \qquad (k \in \mathbb{N}_n)$.

Mit dem Ergibt-Zeichen können wir dafür auch schreiben:

$b := a + b x$ und $c := b + c x$.

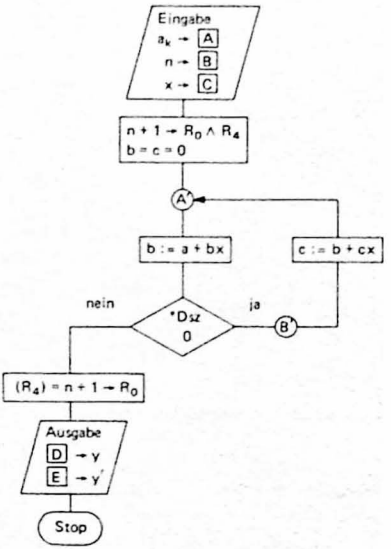
Wir wollen die Werte a_k durch indirekte Speicherung mit [A] in die folgenden Speicher bringen:

$a_n \rightarrow R_{10}$, $a_{n-1} \rightarrow R_{11}$, $a_{n-2} \rightarrow R_{12}$ usw., allgemein $a_{n-j} \rightarrow R_{10+j}$.

[1] s. z.B. Brauch/Dreyer/Haacke: Mathematik für Ingenieure, Teubner-Verlag 1977
oder Collatz/Albrecht: Aufgaben aus der angewandten Mathematik, Vieweg-Verlag, 1972

Im Flußdiagramm und im Programm müssen wir beachten, daß die b-Werte (n + 1)-mal (bis b_0) und die c-Werte n-mal (bis c_1) zu berechnen sind. Wir haben dieses durch die Trennung der Bestimmung der b- und c-Werte durch die Verzweigung erreicht. Das Programm zur Berechnung von y und y' nach dem Horner-Schema soll für dasselbe Polynom und verschiedene x-Werte häufiger benutzt werden, ohne die Koeffizienten a_k und n neu eingeben zu müssen. Dieses erreichen wir, indem wir zu Beginn einer jeden Rechnung $(R_0) = n + 1$ und $b = c = 0$ setzen. Für die Ein- und Ausgabe benutzen wir Programmadreßtasten.

Flußdiagramm:



Programm:

PSS	Taste
000	*LBL
001	A
002	STO *Ind
003	09
004	1
005	SUM
006	9
007	R/S
008	*LBL
009	B
010	+
011	1
012	=
013	STO
014	0
015	STO
016	4
017	R/S
018	*LBL

019	C
020	STO
021	01
022	0
023	STO
024	2
025	STO
026	03
027	9
028	STO
029	9
030	*LBL
031	*A'
032	1
033	SUM
034	9
035	RCL *Ind
036	9
037	+
038	RCL

039	2
040	x
041	RCL
042	1
043	=
044	STO
045	2
046	*Dsz
047	0
048	*B'
049	RCL
050	4
051	STO
052	0
053	R/S
054	*LBL
055	*B'
056	+
057	RCL
058	3

059	x
060	RCL
061	1
062	=
063	STO
064	3
065	*A'
066	*LBL
067	D
068	RCL
069	2
070	R/S
071	*LBL
072	E
073	RCL
074	3
075	R/S
076	
077	
078	

Eingabe: 10 [STO] 09 a_n [A] a_{n-1} [A] ... a_1 [A] a_0 [A] n [B] x [C];

Ausgabe: [D] y [E] y'.

Sollen x, y und y′ über einen Drucker ausgegeben werden, so wird [R/S] in der PSS 053 mit [•Del] gelöscht und ab 068 die nebenstehende Tastenfolge eingegeben.

068	RCL
069	1
070	•Prt
071	RCL
072	2
073	•Prt
074	RCL
075	3
076	•Prt
077	•Adv
078	R/S

Für äquidistante x-Werte mit der Schrittweite h – diesen Wert speichern wir manuell nach R_5 – schreiben wir ab PSS 078:

078	RCL
079	5
080	SUM
081	1
082	GTO
083	0
084	22

und stoppen das Programm manuell mit [R/S].

Beispiel: $y = 1{,}34\,x^4 - 2{,}26\,x^3 + 4{,}88\,x^2 + 3{,}69\,x - 2{,}43$

x	0	0,2	0,4	0,6	0,8	1,0	1,2
y	− 2,430	− 1,513	− 0,284	1,226	3,037	5,220	7,899
y′	3,690	5,414	6,852	8,263	9,903	12,030	14,901

6.3.2. Maximum einer Funktion

$y = f(x)$ sei eine im Intervall $x_0 = a \leqq x \leqq b$ stetige Funktion (s. Bild 6.3.1), deren Kurve im Punkt $P_0(x_0, y_0)$ einen positiven Anstieg besitzen soll. Wir setzen die Existenz eines Maximums dieser Funktion für $\bar{x} > x_0$ voraus: $y_{max} = f(\bar{x})$. Zur Bestimmung von $\bar{x}$ und y_{max} können wir so vorgehen: Wir berechnen die Funktionswerte $y_k = f(x_k)$ für $x_k = x_0 + kh$ $(k \in \mathbb{N}_0)$ solange, bis der nächstfolgende y-Wert kleiner als der vorhergehende wird. Der vorletzte y-Wert ist dann eine Näherung für y_{max}. Die Güte dieses Näherungswertes hängt natürlich von der Schrittweite h ab. Zweckmäßig werden wir h zunächst nicht zu klein wählen, weil sonst unnötig viele Funktionswerte berechnet werden.

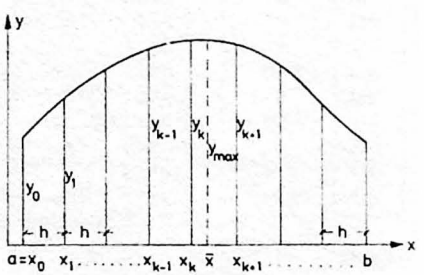


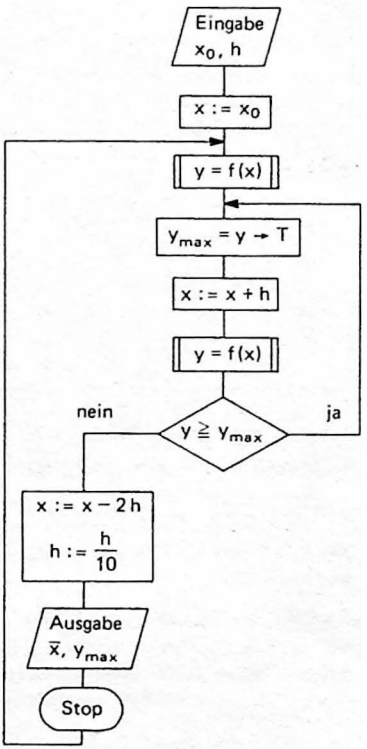
Bild 6.3.1

Gilt für eine Schrittweite h (s. Bild 6.3.1)

$y_{k-1} \leqq y_k$ und $y_{k+1} < y_k$,

so können wir mit $x_0 := x_{k-1}$ und einem kleineren h die Rechnung wiederholen, bis schließlich eine gewünschte Genauigkeit der Werte $\bar{x}$ und y_{max} erreicht worden ist. Wir legen das Programm so an, daß es nach Ermittlung von Näherungswerten für $\bar{x}$ und y_{max} nur mit [R/S] für $x_0 := x_{k-1}$ und $h := \frac{h}{10}$ erneut gestartet werden kann.

Im *Flußdiagramm* haben wir mit y_{max} den jeweils berechneten y-Wert bezeichnet, wenn dieser nicht kleiner als der vorhergehende y-Wert ist. Andernfalls legen wir mit der Anweisung $x := x - 2h$ die neue Ausgangsstelle $x_0 = x_{k-1}$ und mit $h := \frac{h}{10}$ die neue Schrittweite fest.



Das Programm lautet:

PSS	Taste
000	STO
001	0
002	R/S
003	STO
004	1
005	SBR
006	0
007	41
008	x ⇆ t
009	RCL

010	1
011	SUM
012	0
013	SBR
014	0
015	41
016	*x ≧ t
017	0
018	08
019	RCL
020	1

021	+/−
022	SUM
023	0
024	SUM
025	0
026	+/−
027	+
028	RCL
029	0
030	=
031	R/S

032	.
033	1
034	*Prd
035	1
036	x ⇆ t
037	R/S
038	GTO
039	0
040	05
041	

Benutzeranleitung:

Maximum einer Funktion: $y_{max} = f(\bar{x})$

1. Programm eintasten.
2. BAR auf PSS 041 stellen.
3. Programm zur Berechnung von $y = f(x)$ eintasten. $x = (R_0)$. Tastenfolge mit [INV] [SBR] ≙ RTN abschließen und mit [LRN] [RST] in die Betriebsart RECHNEN schalten.
4. Bei Benutzung eines Druckers sind [R/S] in den PSS 031 und 037 durch [*Prt] zu ersetzen.
   Die Rechnung wird manuell gestoppt, wenn die erwünschte Genauigkeit erreicht ist.

Speicherplan		Eingabe	Taste	Anzeige
0	x	x	R/S	–
1	h	h	R/S	$\bar{x}$
T	y_{max}	–	R/S	y_{max}
		–	R/S	$\bar{x}$
		...	usw.	...

Beispiel: $y = f(x) = 1{,}84\,x - x^2 + 1$

Tastenfolge zur Berechnung von $f(x)$:

1.84 [x] [RCL] 0 [−] [RCL] 0 [x^2] [+] 1 [=] . INV SBR

Mit $x_0 = 0$ und $h = 0{,}15$ gibt uns der Drucker die Tabelle 6.3.1 aus. Also (auf 4 Nachkommastellen):

$\bar{x} = 0{,}9200$ und $y_{max} = 1{,}8464$.

Beispiel: Unter allen Flächen, die nach Bild 6.3.2 aus einem Kreis herausgeschnitten werden können, ist diejenige zu bestimmen, die einen größten Umfang besitzt.

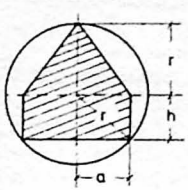


Bild 6.3.2

Zu berechnen sind $x = \frac{a}{r}$ und $y = \frac{h}{r}$ für U_{max}.

$U = 2a + 2h + 2\sqrt{r^2 + a^2}$ und $a^2 + h^2 = r^2$ ergeben mit $a = r\,x$

$$\frac{U}{2r} = f(x) = x + \sqrt{1 - x^2} + \sqrt{1 + x^2} \quad \text{für } 0 \leq x \leq 1.$$

Tastenfolge zur Berechnung von $f(x)$:

[RCL] 0 [+] [(] 1 [−] [RCL] 0 [x^2] [)] [$\sqrt{x}$] [+] [(] 1 [+] [RCL] 0 [x^2] [)] [$\sqrt{x}$] [=] .

Nach Eingabe dieser Tastenfolge starten wir das Programm mit $x_0 = 0$ und $h = 0{,}1$ und erhalten vom Drucker die Tabelle 6.3.2 (mit [*Fix] 7).

Ergebnis: $x = \frac{a}{r} = 0{,}85519$; $\frac{U_{max}}{2r} = 2{,}68931$; $\frac{h}{r} = 0{,}51831$.

(Wer übrigens diese Aufgabe mit Hilfe der Differentialrechnung über $f'(x) = 0$ lösen will, kommt auf eine Gleichung 4. Grades für x^2.)

Tabelle 6.3.1

```
          0.9
        1.846
        0.915
    1.846375
       0.9195
  1.84639975
     0.91995
1.846399998
   0.919995
       1.8464
  0.9199995
       1.8464
 0.92000085
       1.8464
```

Tabelle 6.3.2

```
0.9000000
2.6812523
0.8600000
2.6892330
0.8550000
2.6893122
0.8552000
2.6893123
0.8551900
2.6893124
0.8551890
2.6893124
0.8551886
2.6893124
0.8551885
2.6893124
0.8551885
2.6893124
```

6.3.3. Besselsche Funktionen

In der Technik (z.B. Schwingung einer Membran, Ausbreitung von Wellen) treten *Besselsche Funktionen*[1] auf. Diese durch Differentialgleichungen definierten Funktionen lassen sich durch Potenzreihen oder bestimmte Integrale (s. 6.5.3) darstellen. Wir wollen in diesem Abschnitt die Werte der Besselschen Funktion 0-ter Ordnung berechnen:

$$J_0(x) = 1 - \frac{1}{(1!)^2}\left(\frac{x}{2}\right)^2 + \frac{1}{(2!)^2}\left(\frac{x}{2}\right)^4 - \frac{1}{(3!)^2}\left(\frac{x}{2}\right)^6 + \frac{1}{(4!)^2}\left(\frac{x}{2}\right)^8 - + \ldots$$

Schreiben wir für das n-te Glied ($n \in \mathbb{N}_0$) der Reihe

$a_n = \frac{(-1)^n}{(n!)^2}\left(\frac{x}{2}\right)^{2n}$, so können wir a_n rekursiv aus $a_{n-1} = \frac{(-1)^{n-1}}{((n-1)!)^2}\left(\frac{x}{2}\right)^{2(n-1)}$ berechnen:

$$a_n = -\left(\frac{x}{2n}\right)^2 a_{n-1} \quad \text{und} \quad a_0 = 1.$$

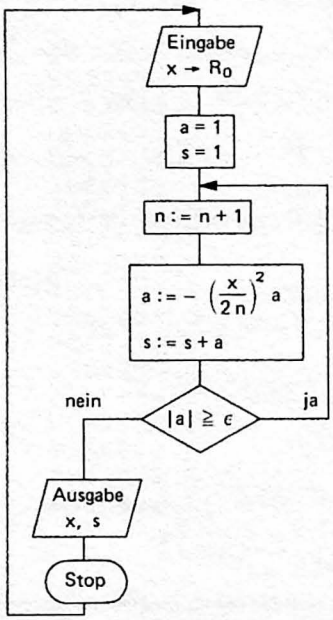
[1] s. Smirnow: Lehrgang der Höheren Mathematik, Teil III, 2, Deutscher Verlag der Wissenschaften, 1971.

Ebenso berechnen wir die Teilsummen s_n der Reihe

$$I_0(x) = s = \sum_{k=0}^{\infty} a_k \quad \text{rekursiv:} \quad s_n = s_{n-1} + a_n \quad \text{und} \quad s_0 = 1.$$

Die Rechnung soll abgebrochen werden, wenn $|a_n| < \epsilon$ wird. Da die Reihe für $I_0(x)$ alternierend (die Glieder besitzen abwechselnde Vorzeichen) mit monoton abnehmenden Gliedern ($|a_n| < |a_{n-1}|$ für hinreichend große n) ist, haben wir damit auch $I_0(x)$ auf eine vorgegebene Genauigkeit ϵ bestimmt.

Flußdiagramm:



Für das folgende Programm geben wir keine Benutzeranleitung. Wir beachten nur die

Eingabe: ϵ [x ⇄ t] , x [R/S]

Danach wird $I_0(x)$ im AR angezeigt.

PSS	Taste
000	*CMs
001	SUM
002	00
003	1
004	STO
005	1
006	STO
007	02
008	1

009	SUM
010	3
011	RCL
012	0
013	÷
014	2
015	÷
016	RCL
017	3
018	=

019	x²
020	+/−
021	*Prd
022	1
023	RCL
024	1
025	SUM
026	2
027	*|x|
028	*x ≧ t

029	0
030	08
031	RCL
032	0
033	R/S
034	RCL
035	2
036	R/S
037	RST

Sollen die Funktionswerte $I_0(x)$ für äquidistante x-Werte mit einer Schrittweite h berechnet werden und steht hierfür ein Drucker zur Verfügung, so ergeben sich die unten links aufgeführten Änderungen im Programm. Mit diesem Programm und $\epsilon = 0{,}000\,000\,1$, $x_0 = 0$ und $h = 0{,}5$ wurden die Werte der Tabelle 6.3.3 ermittelt.

PSS	Taste
000	*Nop
033	*Prt
036	*Prt
037	*Adv
038	0
039	STO
040	3
041	RCL
042	4
043	RST

Eingabe:

ϵ [x ⇄ t]

h [STO] 04

x [R/S]

Tabelle 6.3.3

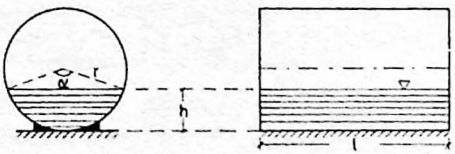
0.000000	5.500000
1.000000	-0.006844
0.500000	6.000000
0.938470	0.150645
1.000000	6.500000
0.765198	0.260095
1.500000	7.000000
0.511828	0.300079
2.000000	7.500000
0.223891	0.266340
2.500000	8.000000
-0.048384	0.171651
3.000000	8.500000
-0.260052	0.041939
3.500000	9.000000
-0.380128	-0.090334
4.000000	9.500000
-0.397150	-0.193929
4.500000	10.000000
-0.320543	-0.245936
5.000000	
-0.177597	

6.4. Nullstellen von Funktionen (Gleichungen)

Unter einer Nullstelle der Funktion $y = f(x)$ oder einer Wurzel (Lösung) der Gleichung $f(x) = 0$ versteht man eine Zahl $\bar{x}$, für die $f(\bar{x}) = 0$ wird. Die Bestimmung einer Nullstelle ist häufig nicht in geschlossener Form, d.h. durch formelmäßiges Auflösen nach $\bar{x}$, möglich. Man ist daher gezwungen, ein Näherungsverfahren anzuwenden, mit dem die Nullstelle beliebig genau berechnet werden kann.

Wir geben zunächst zwei einfache Beispiele für Gleichungen, die zweckmäßig mit einem der weiter unten beschriebenen Verfahren gelöst werden.

Beispiel 1: Ein liegender Zylinder ($r = 1{,}20$ m; $l = 2{,}80$ m) wird mit $V = 4{,}5\ m^3$ Oel gefüllt. Zu bestimmen ist die Höhe h des Oelspiegels.



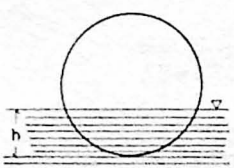
Es gilt (mit α im Bogenmaß):

$$\left(\frac{r^2}{2}\alpha - \frac{r^2}{2}\sin\alpha\right) l = V \quad \text{oder}$$

$$(1)\quad f(\alpha) = \alpha - \sin\alpha - \frac{2V}{r^2 l} = 0 \quad \text{und} \quad h = r\left(1 - \cos\frac{\alpha}{2}\right).$$

Diese (transzendente) Gleichung ist nicht formelmäßig lösbar.

Beispiel 2: Eine Kugel ($r = 246$ mm) aus Kork ($\rho = 0{,}28\ g/cm^3$) schwimmt im Wasser ($\rho_w = 1{,}03\ g/cm^3$). Gesucht ist die Eintauchtiefe h.



Nach dem Archimedischen Prinzip gilt: Masse der Kugel = Masse der verdrängten Wassermenge.

Mit den Formeln für das Volumen einer Kugel und eines Kugelabschnitts wird

$$\rho\frac{4}{3}\pi r^3 = \rho_w \frac{\pi h^2}{3}(3r - h)$$

oder geordnet

$$h^3 - 3rh^2 + 4\frac{\rho}{\rho_w} r^3 = 0.$$

Führen wir die dimensionslose Größe $x = \frac{h}{r}$ ein, so erhalten wir

$$(2)\quad f(x) = x^3 - 3x^2 + 4\frac{\rho}{\rho_w} = 0.$$

Für diese kubische Gleichung gibt es zwar eine Lösungsformel (Cardanosche Formel), doch ist diese im Aufbau so kompliziert, daß sie sich in der Praxis nicht bewährt hat. Wir werden daher auch diese Gleichung mit einem Näherungsverfahren lösen.

6.4.1. Iterationsverfahren

Wir betrachten Gleichungen der Form

$$x = \varphi(x)$$

Die Gleichung $f(x) = 0$ läßt sich z.B. durch $x = x + c\,f(x) = \varphi(x)$ mit einer Konstanten $c \neq 0$ oder auf andere Art stets auf die obige Form bringen.

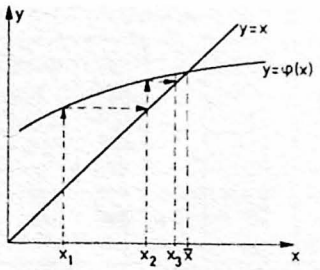


Bild 6.4.1

Graphisch können wir die Lösung der Gleichung $x = \varphi(x)$ als Abszissenwert $\bar{x}$ des Schnittpunkts der Kurven, deren Funktion $y = x$ oder $y = \varphi(x)$ lautet, deuten (s. Bild 6.4.1). Ist x_1 eine Näherung für $\bar{x}$, so wird unter gewissen Bedingungen $x_2 = \varphi(x_1)$ eine bessere Näherung für $\bar{x}$ als x_1 [1]. Wir wollen voraussetzen, daß die Folge $x_1, x_2, x_3, \ldots$, die durch die Iterationsvorschrift

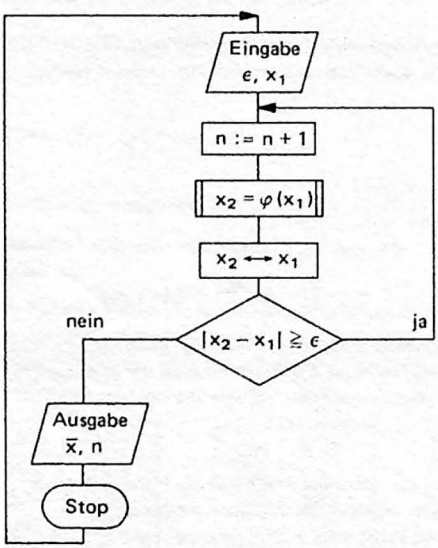
$$x_{n+1} = \varphi(x_n) \quad (n \in \mathbb{N})$$

festgelegt ist, gegen $\bar{x}$ konvergiert.

Bei der Berechnung der Folgenglieder soll solange iteriert werden, bis $|x_{n+1} - x_n| < \epsilon$ wird.

Wir wollen im Programm neben der Berechnung von $\bar{x}$ auch die Anzahl der Iterationsschritte zählen lassen.

Im *Flußdiagramm* setzen wir $x_1 := x_n$ und $x_2 := x_{n+1}$. Die Funktion $\varphi(x)$ soll über ein Unterprogramm aufgerufen werden.



[1] Existenz-, Eindeutigkeits- oder Konvergenzfragen wollen wir hier nicht behandeln. Diese findet der Leser in den Büchern über numerische Mathematik.

Das Programm lautet:

PSS	Taste
000	x ⇆ t
001	R/S
002	STO
003	01
004	0
005	STO
006	02

007	1
008	SUM
009	2
010	SBR
011	0
012	30
013	*Exc
014	1

015	–
016	RCL
017	1
018	=
019	*|x|
020	*x ≧ t
021	0
022	07

023	RCL
024	1
025	R/S
026	RCL
027	2
028	R/S
029	RST
030	

Auf eine Benutzeranleitung verzichten wir. Das Programm ist so kurz und überschaubar, daß jeder sehr schnell erkennt, welche Tasten bei der Eingabe für ϵ und x_1 und zur Berechnung von $\varphi(x)$ über das Unterprogramm zu betätigen sind.

Beispiel 1: Die Gleichung (1) schreiben wir

$$\alpha = \varphi(\alpha) = \sin\alpha + \frac{2V}{r^2 l} .$$

$\frac{2V}{r^2 l} = 2{,}23214$ geben wir in den Speicher R_3.

Die Tastenfolge zur Berechnung von $\varphi(\alpha)$ lautet:

[RCL] 1 [*sin] [+] [RCL] 3 [=] .

Vor der Rechnung ist mit [*Rad] auf das Bogenmaß zu schalten. Wir beginnen die Iteration mit $\alpha_1 = 1{,}8$ ($\approx 100°$, sehr grob geschätzt) und erhalten für einige ϵ-Werte die folgenden Ergebnisse:

ϵ	α	n
0,1	2,7247	21
0,01	2,6742	42
0,001	2,6791	63
0,0001	2,6787	83
0,00001	2,6787	104

$$h = r\left(1 - \cos\frac{\alpha}{2}\right) = 0{,}925\ \text{m} .$$

Das Verfahren konvergiert hier außerordentlich langsam, der Rechner benötigt eine relativ lange Rechenzeit (etwa 145 s für $\epsilon = 0{,}00001$). Aber die Durchführung der Rechnung *ohne* programmierbaren Rechner für z.B. 104 Iterationsschritte würden die meisten von uns wohl als Zumutung ansehen.

Beispiel 2: Die Gleichung (2) können wir auf sehr viele Arten auf die Form $x = \varphi(x)$ bringen. Wir wollen hier gleich das Ergebnis angeben, für das die Iterationsfolge $x_{n+1} = \varphi(x_n)$ konvergiert. Der Leser möge selbst andere ‚Auflösungen nach x' probieren, er wird sicherlich konvergente und divergente Folgen erhalten.

Wir lösen (2) nach x^2 auf und ziehen die Quadratwurzel:

$$x = \varphi(x) = \sqrt{\frac{x^3 + b}{3}} \quad \text{mit} \quad b = \frac{4\rho}{\rho_w} \rightarrow R_3 .$$

Mit der Tastenfolge für $\varphi(x)$

RCL 1 y^x 3 + RCL 3 = ÷ 3 = $\sqrt{x}$

erhalten wir mit $x_1 = 1$:

ϵ	x	n
0,1	0,74560	2
0,01	0,68811	5
0,001	0,68573	7
0,0001	0,68545	9
0,00001	0,68542	11

h = r x = 168,6 mm.

6.4.2. Newtonsches Verfahren

Diesem Verfahren liegt die folgende geometrische Vorstellung zugrunde (s. Bild 6.4.2). Ist x_1 eine Näherung für die gesuchte Nullstelle $\bar{x}$ der Funktion $y = f(x)$, so ersetzen wir die Kurve dieser Funktion durch ihre Tangente im Punkt $P_1(x_1, y_1)$. Der Abszissenwert des Schnittpunkts dieser Tangente mit der x-Achse wird meistens eine bessere Näherung für $\bar{x}$ als x_1 sein. Mit $f'(x_1) = \dfrac{f(x_1)}{x_1 - x_2}$ erhalten wir die Iterationsvorschrift

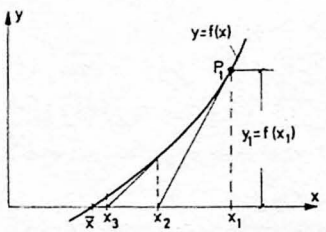


Bild 6.4.2

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} .$$

Wir iterieren wieder bis $|x_{n+1} - x_n| = \left|\dfrac{f(x_n)}{f'(x_n)}\right| < \epsilon$. Die Anzahl n der Iterationsschritte wollen wir ebenfalls berechnen. f(x) und f'(x) werden über ein Unterprogramm eingegeben.

Das Flußdiagramm für diese Aufgabe sieht ganz ähnlich wie beim Iterationsverfahren 6.4.1 aus. Wir wollen deshalb darauf verzichten, es hier aufzuzeichnen, und geben gleich das Programm an. Mit SBR 033 werden f(x) und f'(x) berechnet und nach R_2 bzw. R_3 gespeichert.

PSS	Taste
000	x ⇆ t
001	R/S
002	STO
003	01
004	0
005	STO
006	00
007	1

008	SUM
009	0
010	SBR
011	0
012	33
013	RCL
014	2
015	÷
016	RCL

017	3
018	=
019	+/−
020	SUM
021	1
022	*|x|
023	*x ≧ t
024	0
025	07

026	RCL
027	1
028	R/S
029	RCL
030	0
031	R/S
032	RST
033	

Beispiel 1: Mit

$$f(\alpha) = \alpha - \sin\alpha - \frac{2V}{r^2 l} \quad \text{wird} \quad f'(\alpha) = 1 - \cos\alpha\,.$$

Speichern wir $\frac{2V}{r^2 l}$ nach R_4, so wird die Tastenfolge für das Unterprogramm zur Berechnung von $f(\alpha)$ und $f'(\alpha)$:

RCL 1 − •sin − RCL 4 = STO 02 1 − RCL 1 •cos =
STO 3 .

Das Zahlenbeispiel (Tabelle 6.4.1 mit $\alpha_1 = 1{,}8$) zeigt eine deutliche Verbesserung der Konvergenz gegenüber dem Iterationsverfahren $x_{n+1} = \varphi(x_n)$.

Beispiel 2: Bringen wir $4\,\frac{\rho}{\rho_w}$ in den Speicher R_4, so erhalten wir zur Berechnung von

$$f(x) = x^3 - 3x^2 + 4\frac{\rho}{\rho_w} = x^2(x-3) + 4\frac{\rho}{\rho_w} \quad \text{und} \quad f'(x) = 3x^2 - 6x = 3x\,(x-2)$$

die Tastenfolge

RCL 1 x² × (RCL 1 − 3) + RCL 4 = STO 02 3 × RCL
1 × (RCL 1 − 2) = STO 3 .

Ergebnisse der Rechnung (mit $x_1 = 1$) finden Sie in der Tabelle 6.4.2

Tabelle 6.4.1

ϵ	α	n
0,1	2,678693	3
0,01	2,678693	3
0,001	2,678690	4
0,0001	2,678690	4
0,00001	2,678690	4
0,0000001	2,678690	5

Tabelle 6.4.2

ϵ	x	n
0,1	0,6854519	2
0,01	0,6854154	3
0,001	0,6854154	3
0,0001	0,6854154	3
0,00001	0,6854154	4
0,0000001	0,6854154	4

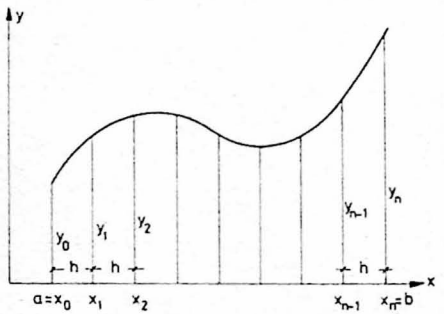
6.5. Numerische Integration

Das bestimmte Integral $I = \int_a^b f(x)\,dx$ wird üblicherweise mit dem Hauptsatz der Differential- und Integralrechnung berechnet. Man sucht eine Stammfunktion $F(x)$ zu $f(x)$, d.h. eine Funktion mit der Eigenschaft $F'(x) = f(x)$, und erhält mit dieser

$$I = \int_a^b f(x)\,dx = F(b) - F(a)\,.$$

Diese Methode ist dann nicht anwendbar, wenn eine Stammfunktion nicht durch bekannte Funktionen angegeben werden kann oder die Funktion $f(x)$ nur in tabellierter Form (z.B. durch Messungen) vorliegt. Auch wenn $F(x)$ sehr schwer zu bestimmen oder nur sehr kompliziert darstellbar ist, wird man I nicht auf die obige Art ermitteln. In allen diesen Fällen hilft ein Verfahren der numerischen Integration weiter.

Wir können uns $I = \int_a^b f(x)\,dx$ als Inhalt der Fläche zwischen der Kurve der Funktion $y = f(x)$ und der x-Achse von $x = a$ bis $x = b$ veranschaulichen. Damit ist die Berechnung des bestimmten Integrals nichts weiter als eine Flächeninhaltsbestimmung. Wir teilen dazu das Intervall [a, b] in n Teilintervalle der Schrittweite $h = \frac{b-a}{n}$ und ersetzen die Kurve durch eine einfachere, für die wir den Flächeninhalt berechnen können. Wählen wir als Ersatzkurve zwischen zwei Punkten der Kurve mit den Abszissenwerten x_{k-1}, x_k jeweils eine Gerade, so erhalten wir die **Trapezregel:**



$$(T)\quad I = \int_a^b f(x)\,dx \cong A_T = \frac{h}{2}(y_0 + y_1) + \frac{h}{2}(y_1 + y_2) + \ldots + \frac{h}{2}(y_{n-1} + y_n)\,.$$

Fassen wir je zwei Teilintervalle zusammen (hierfür muß n gerade gewählt werden) und legen jeweils durch drei Punkte mit den Abszissenwerten x_{k-1}, x_k, x_{k+1} eine Parabel, so führt dieses auf die **Simpsonregel:**

$$(S)\quad I = \int_a^b f(x)\,dx \cong A_S = \frac{h}{3}(y_0 + 4y_1 + y_2) + \frac{h}{3}(y_2 + 4y_3 + y_4) + \ldots + \frac{h}{3}(y_{n-2} + 4y_{n-1} + y_n).$$

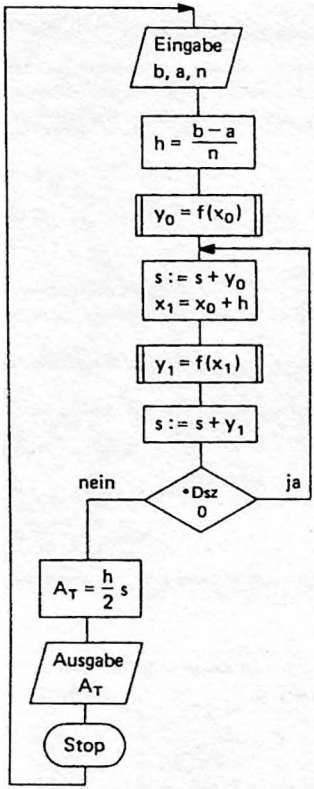
Für diese beiden numerischen Integrationsverfahren (T) und (S) werden wir Programme aufstellen.

6.5.1. Trapezregel

Die in (T) zur Bestimmung von A_T angegebene Reihe aus n Gliedern berechnen wir rekursiv. Die Multiplikation mit $\frac{h}{2}$ führen wir ganz zum Schluß durch:

$s_k = s_{k-1} + (y_{k-1} + y_k)$ und $A_T = \frac{h}{2} s_n$ $(k \in \mathbb{N}_n,\ s_0 = 0)$.

Im *Flußdiagramm* ist $y_0 := y_{k-1}$ und $y_1 := y_k$ gesetzt worden. Im nächsten Intervall wird der im AR stehende Funktionswert $y_1 = y_0$. Durch den Sprung im Programm an die passende Stelle wird erreicht, daß dieser Wert erneut zu s summiert wird. Nur im letzten Intervall — dann ist $(R_0) = 0$ — wird y_1 nur einmal zu s addiert.



Das Programm lautet:

PSS	Taste
000	–
001	R/S
002	STO
003	1
004	=
005	÷
006	R/S
007	STO
008	0
009	=

010	STO
011	2
012	SBR
013	0
014	40
015	SUM
016	3
017	RCL
018	2
019	SUM
020	1

021	SBR
022	0
023	40
024	SUM
025	3
026	•Dsz
027	0
028	0
029	15
030	RCL
031	3

032	×
033	RCL
034	2
035	÷
036	2
037	=
038	R/S
039	RST

Benutzeranleitung:

Numerische Integration: Trapezregel				
1. Programm eintasten. 2. BAR auf PSS 040 stellen. 3. Programm zur Berechnung von f(x) eingeben x = (R_1). Programm mit [INV] [SBR] abschließen, mit [LRN] [RST] in die Betriebsart RECHNEN schalten.				
Speicherplan		Eingabe	Taste	Anzeige
0	n	b	*CM$_s$ R/S	–
1	a, x	a	R/S	–
2	h	n	R/S	A_T
3	s	b	usw.	...

Beispiele zur Trapezregel finden Sie in 6.5.3.

6.5.2. Simpsonregel

Bei der Berechnung des bestimmten Integrals nach (S) werden von den n Teilintervallen stets zwei bei der Berechnung von $\frac{h}{3}(y_{k-1} + 4y_k + y_{k+1})$ zu insgesamt $\frac{n}{2}$ Doppelintervallen zusammengefaßt. Von der geraden Zahl n ist demnach die Hälfte in den Speicher R_0 zu bringen. Wir rechnen wieder rekursiv:

$s := s + (y_0 + 4y_1 + y_2)$ und $A_T = \frac{h}{3} \cdot$ letzter s-Wert.

Auch hier ist der Endwert y_2 in einem Intervall der Anfangswert y_0 im nächsten Intervall. Dieser y-Wert braucht also nur einmal berechnet zu werden. Das Flußdiagramm sieht ähnlich aus wie bei der Trapezregel. Wir wollen daher darauf verzichten, es hier anzugeben. Der Leser möge es sich selbst aufzeichnen.

Das Programm zur Berechnung von A_S wird:

PSS	Taste
000	–
001	R/S
002	STO
003	1
004	=
005	÷
006	R/S
007	STO
008	0
009	=
010	STO
011	02
012	2
013	INV

014	*Prd
015	0
016	SBR
017	0
018	56
019	SUM
020	3
021	RCL
022	2
023	SUM
024	1
025	SBR
026	0
027	56
028	x

029	4
030	=
031	SUM
032	3
033	RCL
034	2
035	SUM
036	1
037	SBR
038	0
039	56
040	SUM
041	3
042	*Dsz
043	0

044	0
045	19
046	RCL
047	3
048	x
049	RCL
050	2
051	÷
052	3
053	=
054	R/S
055	RST

Die Benutzeranleitung gleicht der für die Trapezregel. Lediglich das BAR ist zur Eingabe der Funktion f(x) auf die PSS 056 zu stellen.

6.5.3. Beispiele

Beispiel 1: Es ist das bestimmte Integral $I = \int_1^3 \frac{1}{2x-1}\,dx$ mit der Trapez- und Simpsonregel zu berechnen.

Wir bringen dieses Beispiel, um die Güte der Näherungsverfahren (T) und (S) zu testen.

Wir vergleichen A_T und A_S mit dem exakten Wert $I = \frac{1}{2}\ln(2x-1)\Big|_1^3 = \frac{1}{2}\ln 5 =$

$= 0{,}8047189562$ und geben in der Tabelle neben den Werten A_T und A_S für verschiedene Unterteilungen den prozentualen Fehler $\left|\frac{I-A}{I}\right|$ an.

Die Tastenfolge zur Berechnung von f(x) lautet:

2 [x] [RCL] 1 [−] 1 [=] [1/x] .

n	A_T	Fehler in %	A_S	Fehler in %
2	0,9333333333	16,0	0,8444444444	4,9
6	0,8218004218	2,1	0,8065564066	0,23
10	0,8110195053	0,78	0,8050414970	0,040
20	0,8063124222	0,20	0,8047433945	0,003
100	0,8047829456	0,008	0,8047189986	$5{,}3 \cdot 10^{-6}$

Beispiel 2: Die Funktion f(x) sei tabellarisch gegeben:

x	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
y	0	0,08	0,32	0,65	0,94	1,25	1,44	1,28	0,96	0,61	0

Berechnet werden soll $I = \int_0^1 f(x)\,dx$.

Hier ist die Tastenfolge für f(x) besonders einfach: [R/S] . Der Rechner unterbricht seine Tätigkeit, damit wir den jeweiligen y-Wert eingeben können. Danach starten wir manuell mit [R/S] .

Ergebnis: $A_T = 0{,}753$ und $A_S = 0{,}76$.

Beispiel 3: Die in 6.3.3 durch eine unendliche Reihe festgelegte Besselsche Funktion $J_0(x)$ kann auch durch ein bestimmtes Integral dargestellt werden[1]:

$$J_0(x) = \int_0^1 \cos(x \sin(\pi\varphi))\, d\varphi .$$

Hier ist φ die Integrationsveränderliche, während die Variable x als Parameter unter dem Integral auftritt. Wir wollen $J_0(x)$ für x = 0; 0,5; 1; 1,5; 2 mit der Trapez- und Simpsonregel mit n = 10 berechnen.
Mit $x \rightarrow R_9$ erhalten wir für $f(\varphi) = \cos(x \sin(\pi\varphi))$ die Tastenfolge:

[*π] [x] [RCL] 1 [=] [*sin] [x] [RCL] 9 [=] [*cos] .

Vergessen Sie nicht: Der Winkel φ ist hier im Bogenmaß einzusetzen, also [*Rad] !

x	A_T	A_S
0	1	1
0,5	0,9384698072	0,9384698072
1	0,7651976866	0,7651976864
1,5	0,5118276717	0,5118276619
2	0,2238907791	0,2238906114

6.6. Differentialgleichungen

In diesem Abschnitt werden wir die Anfangswertaufgabe bei gewöhnlichen Differentialgleichungen 1. Ordnung

$y' = f(x, y)$ für $x \geq x_0$ und $y(x_0) = y_0$

behandeln. $f(x, y)$ ist eine gegebene Funktion der beiden Veränderlichen x und y, und x_0 und y_0 sind gegebene Zahlenwerte. Gesucht ist eine Funktion $y(x)$, die Lösung der Differentialgleichung $y' = f(x, y)$ ist und die Anfangsbedingung $y(x_0) = y_0$ erfüllt. Ist diese Lösungsfunktion $y(x)$ nicht in geschlossener Form auffindbar, so muß die obige Anfangswertaufgabe näherungsweise gelöst werden. Hierfür gibt es verschiedene Methoden[2], von denen wir zwei einfache angeben.
Wir bezeichnen mit y_k eine Näherung des gesuchten Funktionswertes $y(x_k) = y(x_0 + kh)$ mit der Schrittweite h und $k \in \mathbb{N}$.

6.6.1. Verbessertes Polygonzugverfahren

$y_1 \approx y(x_1) = y(x_0 + h)$ wird folgendermaßen berechnet:

(VP)
$$y_{1/2} = y_0 + \frac{h}{2} f(x_0, y_0); \qquad x_{1/2} = x_0 + \frac{h}{2};$$
$$y_1 = y_0 + h\, f(x_{1/2}, y_{1/2}); \qquad x_1 = x_{1/2} + \frac{h}{2}.$$

---

1) s. Smirnow: Fußnote 6.3.3

2) s. z.B.
Collatz: Numerische Behandlung von Differentialgleichungen, Springer 1955
Becker/Dreyer/Haacke/Nabert: Numerische Mathematik für Ingenieure, Teubner 1977

Zur Erläuterung diene Bild 6.6.1. Die Strecke von P_0 bis $P_{1/2}$ besitzt den Anstieg $y'_0 = f(x_0, y_0) = f_0$. Die Kurve der Funktion $y(x)$ von x_0 bis x_1 wird ersetzt durch die Strecke P_0P_1 mit dem ‚mittleren' Anstieg $y'_{1/2} = f(x_{1/2}, y_{1/2}) = f_{1/2}$.

Nach Bestimmung von y_1 wird im nächsten Schritt mit den Ausgangswerten x_1, y_1 der Näherungswert y_2 nach derselben Vorschrift (VP) berechnet usw.

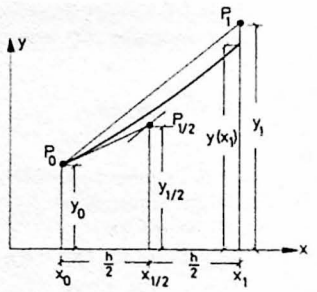
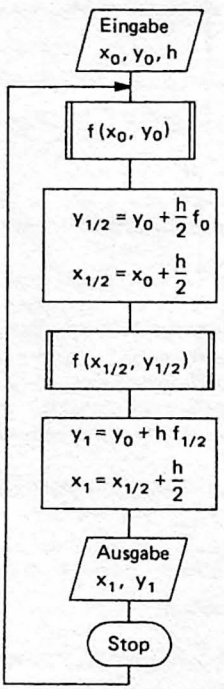


Bild 6.6.1

Das *Flußdiagramm* für den Algorithmus (VP) sieht so aus:



Beim Aufstellen des Programms müssen wir darauf achten, daß bei Beginn des nächsten Schritts die Werte x_1, y_1 in *den* Speichern vorgefunden werden, in denen vorher x_0, y_0 standen.

PSS	Taste
000	STO
001	1
002	R/S
003	STO
004	2
005	STO
006	3
007	R/S
008	STO
009	4
010	÷
011	2

012	=
013	STO
014	5
015	SBR
016	A
017	×
018	RCL
019	5
020	SUM
021	1
022	=
023	SUM
024	2

025	SBR
026	A
027	×
028	RCL
029	4
030	=
031	SUM
032	3
033	RCL
034	5
035	SUM
036	1
037	RCL

038	1
039	R/S
040	RCL
041	3
042	STO
043	2
044	R/S
045	GTO
046	0
047	15
048	*LBL
049	A
050	

Benutzeranleitung:

Verbessertes Polygonzugverfahren

1. Programm eintasten.
2. [GTO] [A] [LRN] betätigen.
3. Programm zur Berechnung von $f(x, y)$ mit $x = (R_1)$ und $y = (R_2)$ eingeben. Mit [INV] [SBR] abschließen und mit [LRN] [RST] in die Betriebsart RECHNEN schalten.

Speicherplan		Eingabe	Taste	Ausgabe
1	$x_0, x_{1/2}, x_1$	x_0	R/S	–
2	$y_0, y_{1/2}, y_1$	y_0	R/S	–
3	y_0, y_1	h	R/S	x_1
4	h	–	R/S	y_1
5	h/2	–	R/S	x_2
		–	R/S	y_2
		–	usw.	...

Ein Beispiel zum verbesserten Polygonzugverfahren finden Sie im Abschnitt 6.6.3.

6.6.2. Differenzenschemaverfahren: Trapezregel

$y(x_1)$ können wir aus $y' = f(x, y)$ formal durch Integration erhalten:

$$y(x_1) = y_0 + \int_{x_0}^{x_1} f(x, y(x))\,dx\,.$$

Lösen wir das Integral näherungsweise mit der Trapezregel (s. 6.5.), so wird

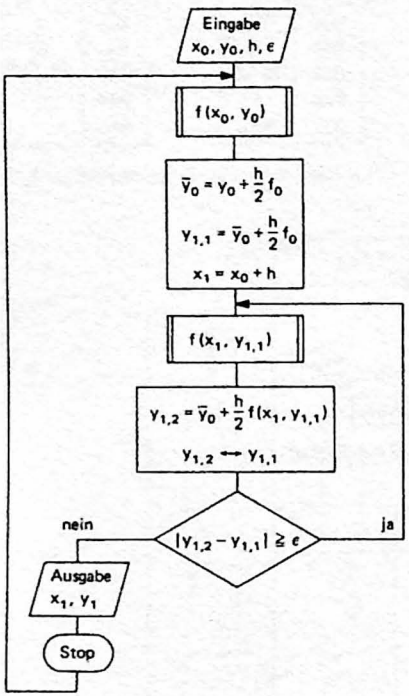
$$y(x_1) \approx y_1 = y_0 + \frac{h}{2}[f(x_0, y_0) + f(x_1, y_1)]\,.$$

Für y_1 erhalten wir eine Gleichung, die wir iterativ lösen:

(DT) $\bar{y}_0 = y_0 + \frac{h}{2} f(x_0, y_0)$; $y_{1,n+1} = \bar{y}_0 + \frac{h}{2} f(x_1, y_{1,n})$.

Als 1. Näherung setzen wir $y_{1,1} = \bar{y}_0 + \frac{h}{2} f(x_0, y_0)$. Iteriert werden soll bis $|y_{1,n+1} - y_{1,n}| < \epsilon$.

Im *Flußdiagramm* setzen wir $y_{1,1} := y_{1,n}$ und $y_{1,2} := y_{1,n+1}$:



PSS	Taste
000	STO
001	1
002	R/S
003	STO
004	2
005	STO
006	3
007	R/S
008	STO
009	4
010	R/S
011	x⇆t
012	SBR
013	E
014	x
015	RCL

016	4
017	SUM
018	1
019	÷
020	2
021	=
022	SUM
023	3
024	+
025	RCL
026	3
027	=
028	STO
029	2
030	SBR
031	E
032	x

033	RCL
034	4
035	÷
036	2
037	+
038	RCL
039	3
040	=
041	*Exc
042	2
043	−
044	RCL
045	2
046	=
047	*|x|
048	*x≧t
049	0

050	30
051	RCL
052	1
053	R/S
054	RCL
055	2
056	STO
057	3
058	R/S
059	GTO
060	0
061	12
062	*LBL
063	E
064	
065	
066	

Benutzeranleitung:

Differenzenschemaverfahren: Trapezregel
1. Programm eintasten. 2. [GTO] [E] [LRN] betätigen. 3. Programm zur Berechnung von f(x, y) mit x = (R_1) und y = (R_2) eingeben. Mit [INV] [SBR] abschließen und mit [LRN] [RST] in die Betriebsart RECHNEN schalten.

Speicherplan		Eingabe	Taste	Anzeige
T	ϵ	x_0	R/S	–
1	x_0, x_1	y_0	R/S	–
2	$y_0 \cdots y_1$	h	R/S	–
3	$y_0, \bar{y}_0, y_1$	ϵ	R/S	x_1
4	h	–	R/S	y_1
		–	R/S	x_2
		–	usw.	...

6.6.3. Beispiel

Bei der Anfangswertaufgabe

$$y' = \sqrt{1+2y}\,(x+e^{-x}) \quad \text{für} \quad x \geqq 0 \quad \text{und} \quad y(0) = \frac{3}{2}$$

vergleichen wir die nach (VP) und (DT) berechneten Näherungswerte mit denen der exakten Lösung

$$y = \frac{1}{2}\left[\left(\frac{x^2}{2} - e^{-x} + 3\right)^2 - 1\right]$$

und geben den prozentualen Fehler $F_k = \left(1 - \frac{y_k}{y(x_k)}\right) 100\,\%$ an.

Die Tastenfolge zur Berechnung von

$$\frac{h}{2} f(x, y) = \frac{h}{2}\sqrt{1+2y}\,(x+e^{-x}) \quad \text{lautet:}$$

1 [+] 2 [×] [RCL] 2 [=] [$\sqrt{x}$] [×] [(] [RCL] 1 [+] [+/−] [INV] [ln x] [)] [=]

Die Ergebnisse der Rechnung mit der Schrittweite h = 0,1 sind in der folgenden Tabelle zusammengefaßt. Der Leser möge sich selbst überlegen, wie das zusätzliche Programm aussehen muß, damit neben y_k auch der exakte Wert $y(x_k)$ und der prozentuale Fehler berechnet werden kann.

x	exakte Lösung y(x)	verb. Polygon-zugverfahren	proz. Fehler	Diff.-schema Trapezregel	proz. Fehler
0	1,5	1,5	0	1,5	0
0,1	1,705341435	1,705190972	0,009	1,705520378	− 0,010
0,2	1,922793149	1,922467494	0,017	1,923174617	− 0,020
0,3	2,154626836	2,154100110	0,024	2,155233906	− 0,028
0,4	2,403278740	2,402523374	0,031	2,404134784	− 0,036
0,5	2,671343909	2,670330159	0,038	2,672473216	− 0,042
0,6	2,961576103	2,960271637	0,044	2,963004367	− 0,048
0,7	3,276891671	3,275261209	0,050	3,278646358	− 0,054
0,8	3,620376098	3,618381104	0,055	3,622486728	− 0,058
0,9	3,995292253	3,992890681	0,060	3,997790605	− 0,063
1,0	4,405089598	4,402235674	0,065	4,408009860	− 0,066

7. Beispiele aus der Technik

7.1. Stereostatik

7.1.1. Kräfte in der Ebene am Punkt

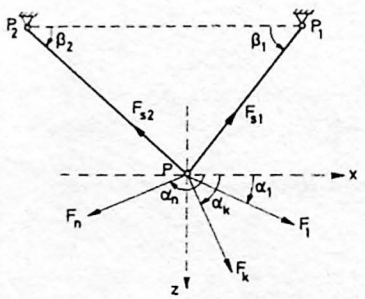


Bild 7.1.1

Zwei Stäbe sind in P gelenkig miteinander verbunden und in P_1 bzw. P_2 gelenkig befestigt (s. Bild 7.1.1). In P greifen n Kräfte $F_1, \ldots F_k, \ldots, F_n$ an, deren Wirkungslinien in der Ebene der beiden Stäbe liegen und mit der x-Achse die Winkel $\alpha_1, \ldots \alpha_k, \ldots, \alpha_n$ bilden. Die Winkel β_1, β_2 und α_k $(k \in \mathbb{N}_n)$ sollen in der eingezeichneten Pfeilrichtung positiv gezählt werden. Berechnet werden sollen die Stabkräfte F_{S1} und F_{S2} [1].

Die Gleichgewichtsbedingung $\Sigma F_x = 0 \wedge \Sigma F_z = 0$ der Statik liefert:

$$F_{S1} \cos\beta_1 - F_{S2} \cos\beta_2 + \sum_{k=1}^{n} F_k \cos\alpha_k = 0 ;$$

$$F_{S1} \sin\beta_1 + F_{S2} \sin\beta_2 - \sum_{k=1}^{n} F_k \sin\alpha_k = 0 .$$

[1] Der Statiker möge mir verzeihen, daß ich hier und in weiteren Abbildungen die Reaktionskräfte in den geometrischen Lageplan hineingezeichnet habe.

Hieraus können wir die Stabkräfte nach folgendem Algorithmus berechnen:

$$s_1 = \sum_{k=1}^{n} F_k \cos\alpha_k ; \qquad s_2 = \sum_{k=1}^{n} F_k \sin\alpha_k ;$$

$$F_{S1} = \frac{s_2 \cos\beta_2 - s_1 \sin\beta_2}{\sin(\beta_1 + \beta_2)} ; \qquad F_{S2} = \frac{F_{S1} \cos\beta_1 + s_1}{\cos\beta_2} .$$

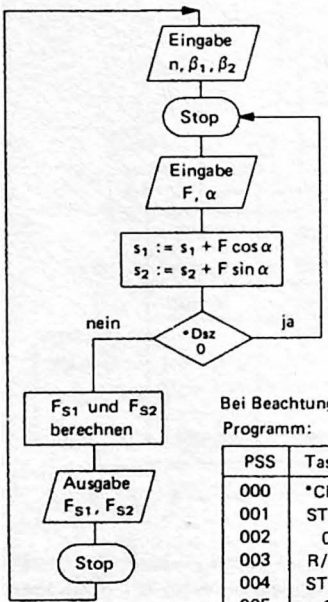
Die Summen s_1 und s_2 berechnen wir rekursiv:

$s_{1k} = s_{1,k-1} + F_k \cos\alpha_k ;$ $\qquad s_{2k} = s_{2,k-1} + F_k \sin\alpha_k$

oder mit der Ergibt-Schreibweise:

$s_1 := s_1 + F\cos\alpha ;$ $\qquad s_2 := s_2 + F\sin\alpha .$

Hiernach entwickeln wir das *Flußdiagramm:*



Bei Beachtung des Speicherplans in der Benutzeranleitung lautet das Programm:

PSS	Taste						
000	•CM,	018	SUM	037	2	056	=
001	STO	019	5	038	•cos	057	R/S
002	0	020	RCL	039	−	058	×
003	R/S	021	3	040	RCL	059	RCL
004	STO	022	×	041	5	060	1
005	1	023	RCL	042	×	061	•cos
006	R/S	024	4	043	RCL	062	+
007	STO	025	•sin	044	2	063	RCL
008	2	026	=	045	•sin	064	5
009	R/S	027	SUM	046	=	065	=
010	STO	028	6	047	÷	066	÷
011	3	029	•Dsz	048	(	067	RCL
012	×	030	0	049	RCL	068	2
013	R/S	031	0	050	1	069	•cos
014	STO	032	09	051	+	070	=
015	4	033	RCL	052	RCL	071	R/S
016	•cos	034	6	053	2	072	RST
017	=	035	×	054	)		
		036	RCL	055	•sin		

Benutzeranleitung:

Berechnung von Stabkräften
1. Programm eintasten. 2. Bei der Eingabe der Winkel Vorzeichen nach Bild 7.1.1 beachten. 3. $F_S > 0$: Zugkraft; $F_S < 0$: Druckkraft.

Speicherplan		Eingabe	Taste	Anzeige
0	n	n	R/S	–
1	β_1	β_1	R/S	–
2	β_2	β_2	R/S	–
3	F	F_1	R/S	–
4	α	α_1	R/S	–
5	s_1	F_2	R/S	–
6	s_2	...	...	...
		α_n	R/S	F_{S1}
		–	R/S	F_{S2}

In den folgenden Beispielen dient die erste Zeile als Test.

n	β_1	β_2	F_k/kN	α_k	F_{S1}/kN	F_{S2}/kN
1	30°	30°	5	90°	5,000	5,000
1	48,6°	34,5°	8,40	60°	3,643	8,019
1	106°	32°	6,32	110°	9,239	–5,552
5	40°	30°	4,65	0°		
			3,08	45°		
			6,72	90°		
			5,26	135°		
			2,60	180°	11,358	10,634

7.1.2. Auflagerkräfte beim Balken

Ein gelenkig gelagerter Balken werde nach Bild 7.1.2 mit n Kräften belastet. Zur Berechnung der Auflagerkräfte F_B, F_{Az}, F_{Ax} wollen wir ein Programm aufstellen. Eingegeben werden n, l, Δx_k, F_k, α_k ($k \in \mathbb{N}_n$). Δx_1 wird negativ eingegeben, falls F_1 links vom Auflager A liegt. Alle weiteren Δx_k sind stets positiv. Die Vorzeichenfestsetzungen für α_k und F_k sind aus Bild 7.1.2 ersichtlich.

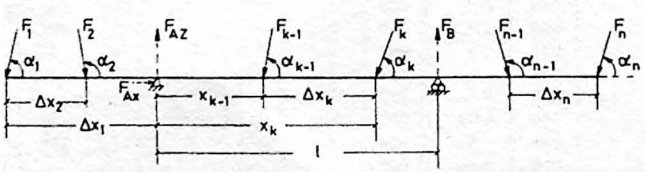


Bild 7.1.2

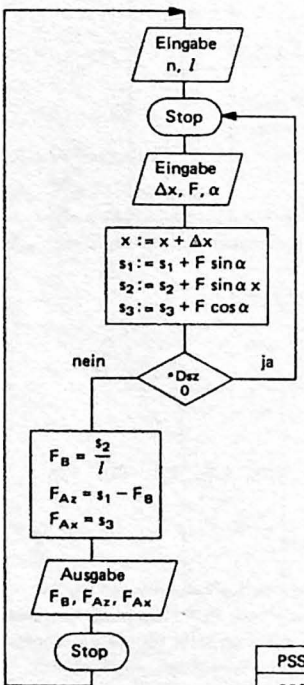
Aus der Gleichgewichtsbedingung der Statik folgt:

$$x_k = x_{k-1} + \Delta x_k; \quad F_B\, l = \sum_{k=1}^{n} F_k \sin\alpha_k\, x_k = s_2;$$

$$F_{Az} = \sum_{k=1}^{n} F_k \sin\alpha_k - F_B = s_1 - F_B; \quad F_{Ax} = \sum_{k=1}^{n} F_k \cos\alpha_k = s_3.$$

Alle auftretenden Summen berechnen wir wie in 7.1.1. rekursiv.

Flußdiagramm:



PSS	Taste
000	•CM$_s$
001	STO
002	0
003	R/S
004	STO
005	1
006	R/S
007	SUM
008	2
009	R/S
010	STO
011	3
012	x
013	R/S

014	STO
015	4
016	•sin
017	=
018	SUM
019	5
020	x
021	RCL
022	2
023	=
024	SUM
025	6
026	RCL
027	3
028	x

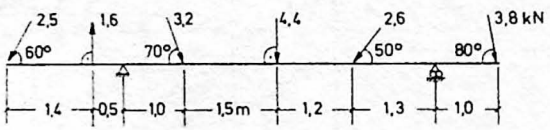
029	RCL
030	4
031	•cos
032	=
033	SUM
034	7
035	•Dsz
036	0
037	0
038	06
039	RCL
040	6
041	÷
042	RCL
043	1

044	=
045	R/S
046	+/−
047	+
048	RCL
049	5
050	=
051	R/S
052	RCL
053	7
054	R/S
055	RST

Benutzeranleitung:

Auflagerkräfte beim Balken				
1. Programm eintasten. 2. Liegt F_1 links vom Auflager A, so sind Δx_1 negativ, alle weiteren Δx_k stets positiv einzugeben. Für α_k und F_k sind die Vorzeichenfestsetzungen nach Bild 7.1.2 zu beachten.				
Speicherplan		Eingabe	Taste	Anzeige
0	n	n	R/S	–
1	l	l	R/S	–
2	x	Δx_1	R/S	–
3	F	F_1	R/S	–
4	α	α_1	R/S	–
5	s_1	Δx_2	R/S	–
6	s_2	...	...	...
7	s_3	α_n	R/S	F_B
		–	R/S	F_{Az}
		–	R/S	F_{Ax}

Beispiel:



Hier werden die folgenden Zahlenwerte eingegeben:

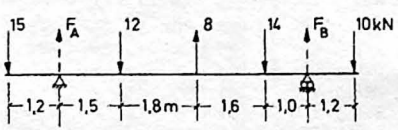
6; 5,0; −1,9; 2,5; 60; 1,4; −1,6; 90; 1,5; 3,2; 110; 1,5; 4,4; 90; 1,2; 2,6; 50; 2,3; 3,8; 100

Ergebnis: $F_B = 8{,}10\,\text{kN}$; $F_{Az} = 5{,}60\,\text{kN}$; $F_{Ax} = 1{,}17\,\text{kN}$.

Für den Sonderfall, daß *alle* Kräfte *senkrecht* zur Balkenachse stehen, könnten wir im obigen Programm jedesmal $\alpha = 90°$ eingeben. Dieses werden wir sicherlich auch tun, wenn das Programm bereits im Rechner gespeichert ist. Haben wir aber öfter für diesen Sonderfall, der in der Praxis häufig vorkommt, die Auflagerkräfte zu berechnen, so empfiehlt sich die Eingabe eines einfacheren Programms. In diesem Fall gilt:

$$x_k = x_{k-1} + \Delta x_k; \quad F_B\, l = \sum_{k=1}^{n} F_k\, x_k; \quad F_{Az} = \sum_{k=1}^{n} F_k - F_B; \quad F_{Ax} = 0.$$

Der Leser möge hierfür selbst ein Programm aufstellen (s. auch die Bestimmung von $M_{b\max}$ in 7.2) und mit diesem das folgende Beispiel durchrechnen.



Ergebnis:

$F_A = 23{,}8\,\text{kN}$; $F_B = 19{,}2\,\text{kN}$.

7.2. Elastostatik (Festigkeitslehre)

7.2.1. Flächenmomente

Für eine Fläche, die wir uns aus n Teilflächen zusammengesetzt denken (s. Bild 7.2.1) bezeichnet man die folgenden Größen als Flächenmomente[1] 0., 1. oder 2. Ordnung:

$$A = \sum_{k=1}^{n} A_k \qquad \text{(Flächeninhalt)};$$

$$A z_s = \sum_{k=1}^{n} A_k z_k \qquad \text{(statisches Moment der Fläche, } z_s = \text{Abstand des Schwerpunktes von der y-Achse)};$$

$$I_y = \sum_{k=1}^{n} (A_k z_k^2 + I_{sk}) \qquad \text{(Flächenträgheitsmoment, bezogen auf die y-Achse)};$$

$$I_s = I_y - A z_s^2 \qquad \text{(Flächenträgheitsmoment, bezogen auf die Schwerachse; Steinerscher Satz)}.$$

Hierin bedeuten

A_k den Flächeninhalt
z_k die Schwerpunktskoordinate
I_{sk} das Eigenflächenträgheitsmoment

} der k-ten Fläche.

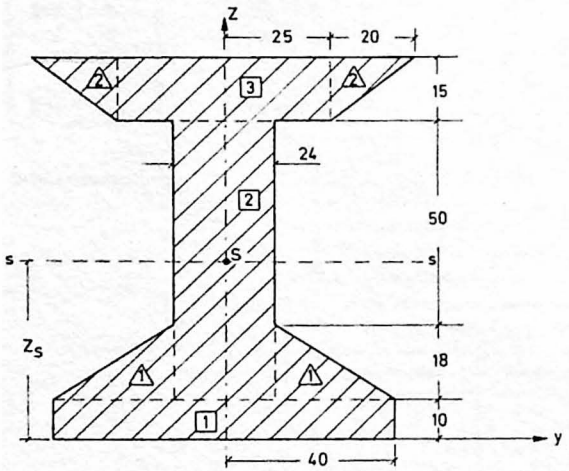


Bild 7.2.1

Wir wollen annehmen, die Fläche bestehe aus n_1 Rechtecken der Breite b_k und der Höhe h_k und n_2 Dreiecken mit der Grundseite $\bar{b}_k$ und der Höhe $\bar{h}_k$ (s. Bild 7.2.1). Hierfür gelten:

$$A_k = b_k h_k \quad \text{und} \quad I_{sk} = \frac{b_k h_k^3}{12} = \frac{A_k h_k^2}{12} \qquad \text{für das Rechteck,}$$

$$\bar{A}_k = \frac{1}{2} \bar{b}_k \bar{h}_k \quad \text{und} \quad I_{sk} = \frac{\bar{b}_k \bar{h}_k^3}{36} = \frac{\bar{A}_k \bar{h}_k^2}{18} \qquad \text{für das Dreieck.}$$

[1] s. z.B. Holzmann/Meyer/Schumpich: Technische Mechanik, Teil 3, Teubner 1975

Zusammenstellung aller Formeln ergibt:

$$A = s_1 = \sum_{k=1}^{n_1} b_k h_k + \frac{1}{2} \sum_{k=1}^{n_2} \bar{b}_k \bar{h}_k ; \quad A z_s = s_2 = \sum_{k=1}^{n_1} A_k z_k + \sum_{k=1}^{n_2} \bar{A}_k \bar{z}_k ;$$

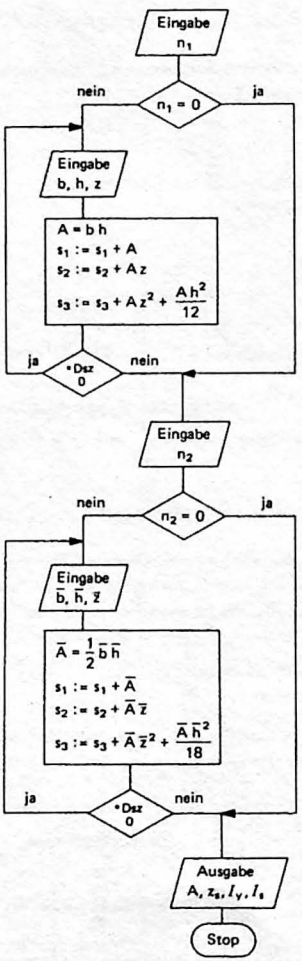
$$I_y = s_3 = \sum_{k=1}^{n_1} \left(A_k z_k^2 + \frac{A_k h_k^2}{12} \right) + \sum_{k=1}^{n_2} \left(\bar{A}_k \bar{z}_k^2 + \frac{\bar{A}_k \bar{h}_k^2}{18} \right) ; \quad I_s = I_y - A z_s^2 .$$

Zur Berechnung dieser Werte A, z_s, I_y, I_s schreiben wir ein Programm. Die Zahlenwerte

$n_1, b_1, h_1, z_1, b_2, \ldots, z_{n1}$; $n_2, \bar{b}_1, \bar{h}_1, \bar{z}_1, \bar{b}_2, \ldots, \bar{z}_{n2}$

werden in dieser Reihenfolge eingegeben.

Flußdiagramm:



Die Berechnung der Summen in den beiden Zweigen für n_1 und n_2 weisen viele Gemeinsamkeiten auf: Von b h; s_1 = ... bis ... $\overline{A\,h^2}$ werden dieselben Programmschritte benutzt. Wir fassen sie daher zu einem Unterprogramm (SBR 061) zusammen. Auf diese Weise können viele (genau: 16) Programmspeicherplätze gespart werden.

PSS	Taste
000	*CM$_s$
001	*CP
002	STO
003	0
004	*x = t
005	0
006	19
007	SBR
008	0
009	61
010	1
011	2
012	=
013	SUM
014	3
015	*Dsz
016	0
017	0
018	07
019	R/S
020	STO
021	0

022	*x = t
023	0
024	40
025	.
026	5
027	×
028	SBR
029	0
030	61
031	1
032	8
033	=
034	SUM
035	3
036	*Dsz
037	0
038	0
039	25
040	RCL
041	2
042	÷
043	RCL
044	1

045	R/S
046	=
047	R/S
048	x^2
049	×
050	RCL
051	1
052	−
053	RCL
054	3
055	R/S
056	=
057	+/−
058	*Nop
059	R/S
060	RST
061	R/S
062	×
063	R/S
064	STO
065	4
066	=
067	SUM

068	1
069	STO
070	6
071	×
072	R/S
073	STO
074	5
075	=
076	SUM
077	2
078	×
079	RCL
080	5
081	+
082	RCL
083	6
084	×
085	RCL
086	4
087	x^2
088	÷
089	INV SBR

Benutzeranleitung:

Berechnung von Flächenmomenten

1. Programm eintasten.
2. Für negative Flächenteile wird b negativ, h positiv eingegeben.
3. Bei Benutzung eines Druckers ist in die PSS 045, 047, 055, 058 die Anweisung *Prt zu setzen.

Speicherplan		Eingabe	Taste	Anzeige
0	n	n_1	R/S	−
1	s_1	b_1	R/S	−
2	s_2	h_1	R/S	−
3	s_3	z_1	R/S	−
4	h	...	...	...
5	z	z_{n1}	R/S	−
6	A_k	n_2	R/S	−
		$\overline{b}_1$	R/S	−
		...	...	...
		$\overline{z}_{n2}$	R/S	A
		−	R/S	z_s
		−	R/S	I_y
		−	R/S	I_s

Beispiel: Wir berechnen A, z_s, I_y und I_s für die in Bild 7.2.1 dargestellte Fläche. Für die Eingabe stellen wir die folgende Tabelle zusammen:

n	Fläche	b/cm	h/cm	z/cm
3	1	8	1	0,5
	2	2,4	6,8	4,4
	3	5	1,5	8,55
2	1	5,6	1,8	1,6
	2	4	1,5	8,8

Ergebnis:

$A = 39{,}86 \quad cm^2;$
$z_s = 43{,}8 \quad mm;$
$I_y = 1177{,}7 \quad cm^4;$
$I_s = 414{,}7 \quad cm^4.$

7.2.2. Maximales Biegemoment

Ein an den Enden gelenkig gelagerter Balken werde nach Bild 7.2.2 mit n Kräften $F_1, \ldots, F_k, \ldots, F_n$ senkrecht zur Balkenachse belastet. Es soll ein Programm zur Ermittlung der Auflagerkräfte F_A und F_B, des maximalen Biegemoments $M_{b\,max} = |M_b|_{max}$ und der Stelle $\bar{x}$, an der $M_{b\,max}$ angenommen wird, geschrieben werden.

Für die Berechnung der Auflagerkräfte hatten wir bereits in 7.1.2. ein Programm aufgestellt. Dort hatten wir nach der Eingabe der Werte Δx_k, F_k, α_k diese sofort verarbeitet und mit der Eingabe der nächsten Werte gelöscht. Hier gehen wir anders vor, weil wir die Δx_k und F_k zunächst zur Berechnung der Auflagerkräfte F_A und F_B benötigen und danach zur Berechnung der Querkräfte F_{Qk} und der Biegemomente M_{bk}.

Für die Berechnung benutzen wir die aus Bild 7.2.2 ersichtlichen Formeln:

$$F_B\, l = \Sigma F_k x_k; \quad F_A = \Sigma F_k - F_B;$$

$$F_{Qk} = F_{Q,k-1} - F_{k-1}; \quad F_{Q0} = 0; \quad F_0 = -F_A;$$

$$M_{bk} = M_{b,k-1} + F_{Qk}\,\Delta x_k.$$

Das Biegemoment wird über die Querkraftfläche berechnet. Positive Werte F_Q oder M_b werden nach unten, negative nach oben (s. Bild 7.2.2) aufgetragen.

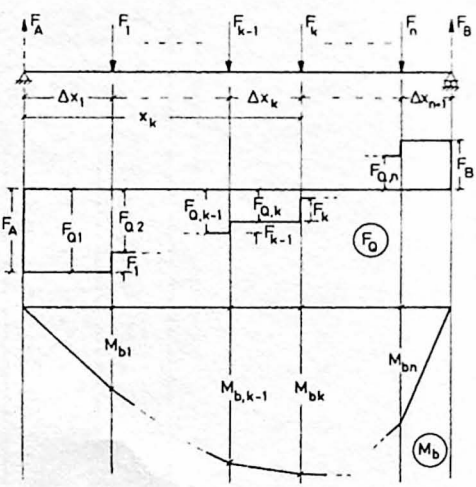
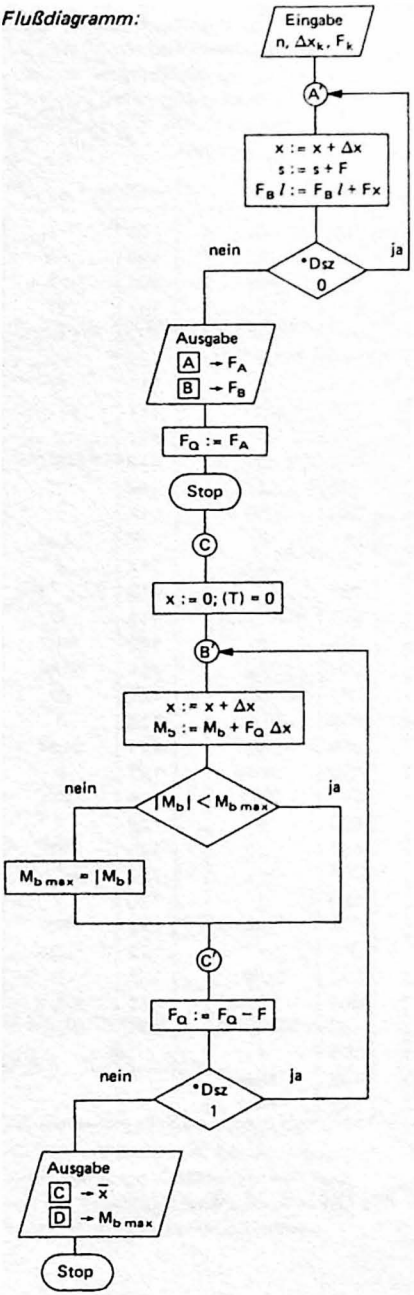


Bild 7.2.2

Flußdiagramm:



Speicherplan	
T	$M_{b\,max}$
0	n, ..., 0
1	n, ..., 0
2	F_A, F_Q
3	F_B
4	M_b
5	x
6	Ind
7	Δx_1; $\bar{x}$
8	F_1
9	Δx_2
10	F_2
11	Δx_3
12	usw.

Das Programm schreiben wir so, daß wahlweise $\bar{x}$ und $M_{b\,max}$ oder x_k, F_{Qk} und M_{bk} für $k \in \mathbb{N}_n$ ausgegeben werden (z.B. über einen Drucker). Hierfür dienen die [*Nop]-Anweisungen und [*Exc] 5. Für die Eingabe der Werte Δx_k, F_k und deren Aufruf im Programm wird die indirekte Speicheradressierung benutzt. Das Programm zur Berechnung von F_A und F_B ist in der PSS 071 abgeschlossen. In dem ab PSS 075 beginnenden Programmteil werden $\bar{x}$ und $M_{b\,max}$ bzw. F_Q und M_b ermittelt.

PSS	Taste
000	*CM$_s$
001	STO
002	0
003	+
004	1
005	=
006	STO
007	01
008	7
009	STO
010	6
011	R/S
012	*LBL
013	E
014	STO *Ind
015	06
016	1
017	SUM
018	6
019	R/S
020	*LBL
021	A
022	7
023	STO
024	6
025	*LBL
026	*A′
027	RCL *Ind
028	6
029	SUM
030	05
031	1
032	SUM
033	6

034	RCL *Ind
035	6
036	SUM
037	2
038	x
039	RCL
040	5
041	=
042	SUM
043	03
044	1
045	SUM
046	6
047	*Dsz
048	0
049	*A′
050	RCL *Ind
051	6
052	SUM
053	5
054	RCL
055	5
056	INV
057	*Prd
058	3
059	RCL
060	3
061	INV
062	SUM
063	2
064	RCL
065	2
066	R/S
067	*LBL
068	B

069	RCL
070	3
071	R/S
072	*Nop
073	*LBL
074	C
075	0
076	STO
077	5
078	*CP
079	7
080	STO
081	6
082	*LBL
083	*B′
084	RCL *Ind
085	6
086	SUM
087	5
088	*Exc
089	5
090	*Nop
091	*Exc
092	5
093	x
094	RCL
095	2
096	*Nop
097	=
098	SUM
099	4
100	RCL
101	4
102	*Nop
103	*Nop

104	*|x|
105	INV
106	*x ≧ t
107	*C′
108	x ↘ t
109	RCL
110	5
111	STO
112	7
113	*LBL
114	*C′
115	1
116	SUM
117	6
118	RCL *Ind
119	6
120	INV
121	SUM
122	02
123	1
124	SUM
125	6
126	*Dsz
127	1
128	*B′
129	RCL
130	7
131	R/S
132	*LBL
133	D
134	x ↘ t
135	R/S

Eingabe: [RST] n [R/S] Δx_1 [E] F_1 [E] Δx_2 [E] ... F_n [E] Δx_{n+1} [E];

Ausgabe: [A] F_A [B] F_B [C] $\bar{x}$ [D] $M_{b\,max}$.

Beispiel:

Für die Zahlenwerte der nebenstehenden Tabelle erhalten wir die Ergebnisse:

$F_A = 14{,}21$ kN;
$F_B = 20{,}39$ kN;
$\bar{x} = 4{,}9$ m;
$M_{b\,max} = 34{,}59$ kN m.

Δx/m	F/kN
1,4	8,4
1,5	6,9
1,2	-10,2
0,8	9,4
1,0	11,3
1,2	8,8
1,0	

Benutzung des Druckers PC 100A: [*Prt] in die PSS 066, 071, 090, 096, 102 und [*Adv] in 072, 103 setzen. Mit diesem Programm und [*Fix] 2 erhalten wir die Tabelle 7.2.1.

Tabelle 7.2.1

```
14.21        4.10         7.10
20.39       -1.09       -11.59
            27.30        20.39

 1.40
14.21        4.90         8.10
19.89        9.11       -20.39
            34.59         0.00

 2.90
 5.81        5.90
28.61       -0.29
            34.30
```

7.3. Kinematik/Kinetik

7.3.1. Kurbeltrieb

Für den in Bild 7.3.1 dargestellten Kurbeltrieb wollen wir zur Berechnung des Weges s, der Geschwindigkeit v und der Beschleunigung a des Kreuzkopfes K ein Programm schreiben.

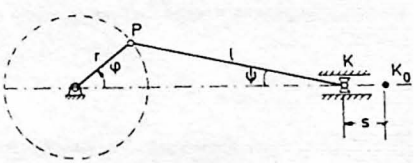


Bild 7.3.1

Den Weg s messen wir von der rechten Totlage K_0 des Kreuzkopfes. Nennen wir $\lambda = \frac{r}{l}$ das Verhältnis der Kurbelstange zur Schubstange, so berechnen wir für eine konstante Drehzahl n die gesuchten Größen nach folgendem Rechenschema[1]:

$$\psi = \arcsin(\lambda \sin\varphi);$$

$$\frac{s}{r} = \bar{s} = 1 + \frac{1}{\lambda} - \cos\varphi - \frac{1}{\lambda}\cos\psi;$$

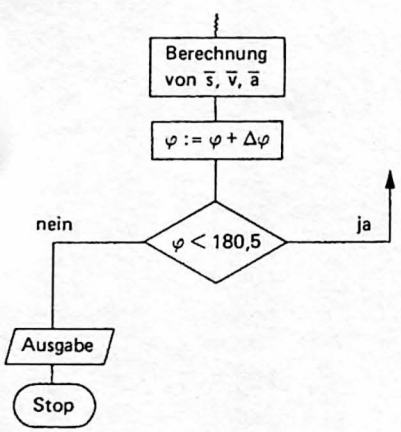
$$\frac{v}{r\omega} = \bar{v} = \sin\varphi + \frac{\lambda}{2}\frac{\sin 2\varphi}{\cos\psi};$$

$$\frac{a}{r\omega^2} = \bar{a} = \frac{1}{\cos\psi}\left[\cos(\varphi+\psi) + \lambda\left(\frac{\cos\varphi}{\cos\psi}\right)^2\right].$$

Hierin bedeuten $\omega = 2\pi n$ die Winkelgeschwindigkeit der Kurbelstange, $r\omega$ die Umfangsgeschwindigkeit und $r\omega^2$ die Zentripetalbeschleunigung des Gelenkpunktes P.

$\bar{s}$, $\bar{v}$, $\bar{a}$ sollen von $\varphi = 0°$ mit einer Schrittweite $\Delta\varphi$ bis $\varphi = 180°$ berechnet und mit dem zugehörigen Winkel φ über den Drucker ausgegeben werden. Nach der Ausgabe für $\varphi = 180°$ soll der Rechner selbständig stoppen.

Der wesentliche Teil des *Flußdiagramms* sieht folgendermaßen aus:



Den Vergleich haben wir mit 180,5 statt mit 180 durchgeführt, da sonst die letzten Werte für $\varphi = 180 - \Delta\varphi$ berechnet und ausgegeben werden. Die Abfrage $180 < 180$ wird verneint, dagegen $180 < 180{,}5$ bejaht. (Ist $\Delta\varphi \leqq 0{,}5°$, so dürfen wir natürlich nicht 180,5 wählen. Wir gehen auf jeden Fall sicher, wenn wir den Vergleich $\varphi < 180 + \frac{\Delta\varphi}{2}$ durchführen.)

---

[1] Der mathematisch Interessierte möge diese Beziehungen aus der Geometrie und mit $v = \frac{ds}{dt} = \frac{ds}{d\varphi}\omega$ und $a = \frac{dv}{dt} = \frac{dv}{d\varphi}\omega$ bestätigen.

Speicherplan	
T	180,5
0	λ
1	$\Delta\varphi$
2	φ
3	ψ
4	$\cos\psi$

Mit dem Speicherplan lautet das Programm:

PSS	Taste
000	*CM_s
001	STO
002	0
003	R/S
004	STO
005	1
006	RCL
007	2
008	*Prt
009	*sin
010	×
011	RCL
012	0
013	=
014	INV
015	*sin
016	STO
017	3
018	*cos
019	STO
020	4
021	÷
022	RCL
023	0

024	+
025	RCL
026	2
027	*cos
028	−
029	RCL
030	0
031	1/x
032	−
033	1
034	=
035	+/−
036	*Prt
037	RCL
038	2
039	*sin
040	+
041	RCL
042	0
043	÷
044	2
045	×
046	(
047	RCL
048	2

049	×
050	2
051	)
052	*sin
053	÷
054	RCL
055	4
056	=
057	*Prt
058	RCL
059	2
060	+
061	RCL
062	3
063	=
064	*cos
065	+
066	(
067	RCL
068	2
069	*cos
070	÷
071	RCL
072	3
073	*cos

074	)
075	x^2
076	×
077	RCL
078	0
079	=
080	÷
081	RCL
082	4
083	=
084	*Prt
085	*Adv
086	RCL
087	1
088	SUM
089	2
090	RCL
091	2
092	INV
093	*x ≧ t
094	0
095	06
096	R/S

Eingabe: 180,5 [x ⇄ t] λ [R/S] Δφ [R/S] .

Mit $\lambda = \frac{1}{3{,}8}$ und $\Delta\varphi = 15°$ gibt uns der Drucker die Tabelle 7.3.1 (mit [•Fix] 5 auf 5 Nachkommastellen) für $\varphi, \bar{s}, \bar{v}, \bar{a}$ aus:

Tabelle 7.3.1

0.00000	105.00000
0.00000	1.38363
0.00000	0.89790
1.26316	-0.49320
15.00000	120.00000
0.04290	1.60000
0.32476	0.74899
1.19550	-0.63143
30.00000	135.00000
0.16701	1.77348
0.61495	0.57319
1.00227	-0.70230
45.00000	150.00000
0.35926	1.89906
0.84102	0.38505
0.71191	-0.72978
60.00000	165.00000
0.60000	1.97475
0.98306	0.19288
0.36857	-0.73635
75.00000	180.00000
0.86600	2.00000
1.03395	0.00000
0.02444	-0.73684
90.00000	
1.13394	
1.00000	
-0.27277	

7.3.2. Bahnkurve

Ein Punkt Q bewege sich nach Bild 7.3.2 auf einer Geraden g und werde von einem Punkt P verfolgt[1]. Beide Punkte bewegen sich mit konstanter Geschwindigkeit v_P bzw. v_Q und befinden sich zur Zeit $t = 0$ in P_0 bzw. Q_0. Bei der Verfolgung bewegt sich P in jedem Augenblick in Richtung auf den Punkt Q. Wie lange dauert es, bis P von Q einen Abstand besitzt, der kleiner als ϵ ist? Wo befindet sich dann der Punkt P, und welchen Weg hat er zurückgelegt?

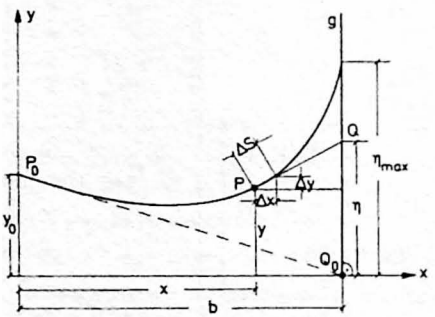


Bild 7.3.2

Wir werden das Programm so schreiben, daß es in zweierlei Form benutzt werden kann.

1. Wir wollen zu jeder Zeit t die Koordinaten $x = x(t)$, $y = y(t)$ des Punktes P der Bahnkurve und den Abstand $d = |PQ|$ vom Drucker ausgeben lassen.
2. Es sollen nur die Zeit t, die Koordinaten x, y, η und der von P zurückgelegte Weg s ausgegeben werden, sobald $d = |PQ| < \epsilon$ erreicht ist.

Zunächst zur mathematischen Formulierung des Problems. Zur Zeit t besitze P die Koordinaten x, y und Q die Ordinate η. Der Abstand der beiden Punkte beträgt

$$d = \sqrt{(b-x)^2 + (\eta - y)^2} .$$

In der Zeit Δt legt Q den Weg $\Delta\eta = v_Q \Delta t$ und P $\Delta s = v_P \Delta t$ in Richtung auf Q zurück. Bei Anwendung des Strahlensatzes erhalten wir:

$$\Delta x = \frac{\Delta s}{d}(b-x) \quad \text{und} \quad \Delta y = \frac{\Delta s}{d}(\eta - y) .$$

Die neue Position der Punkte P und Q zur Zeit $t := t + \Delta t$ wird:

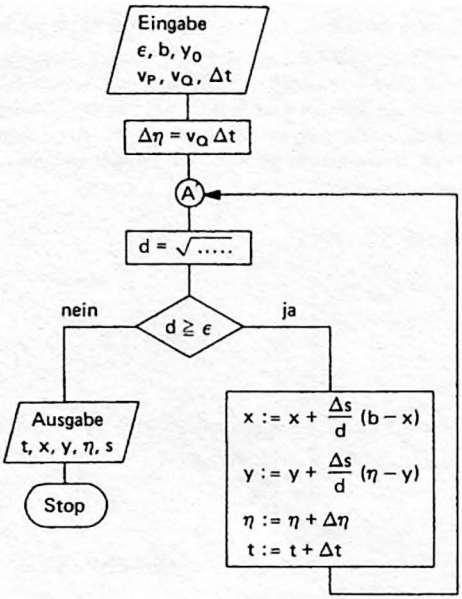
$$x := x + \Delta x, \quad y := y + \Delta y, \quad \eta := \eta + \Delta\eta .$$

Der von P in der Zeit t zurückgelegte Weg beträgt: $s = v_P t$.

---

[1] Wer dieses Problem lieber als „Aufgabe aus der Praxis" formuliert haben möchte, möge sich P als Hund vorstellen, der eine Wurst Q in der Hand seines Herrn „verfolgt".

Flußdiagramm:



Das Programm lautet:

PSS	Taste
000	*CM$_x$
001	STO
002	4
003	R/S
004	STO
005	7
006	R/S
007	STO
008	2
009	R/S
010	STO
011	3
012	R/S
013	STO
014	0
015	*Prd
016	3
017	*LBL
018	*A'
019	RCL
020	1
021	*Nop
022	RCL
023	4
024	−
025	RCL
026	6
027	*Nop
028	=

029	x^2
030	+
031	(
032	RCL
033	5
034	−
035	RCL
036	7
037	*Nop
038	)
039	x^2
040	=
041	$\sqrt{x}$
042	*Nop
043	*Nop
044	*$x \geqq t$
045	0
046	49
047	0
048	R/S
049	1/x
050	×
051	RCL
052	2
053	×
054	RCL
055	0
056	=
057	STO
058	8

059	×
060	(
061	RCL
062	4
063	−
064	RCL
065	6
066	)
067	=
068	SUM
069	6
070	RCL
071	5
072	−
073	RCL
074	7
075	=
076	×
077	RCL
078	8
079	=
080	SUM
081	7
082	RCL
083	3
084	SUM
085	5
086	RCL
087	0
088	SUM

089	1
090	*A'
091	*LBL
092	A
093	RCL
094	1
095	R/S
096	*LBL
097	B
098	RCL
099	6
100	R/S
101	*LBL
102	C
103	RCL
104	7
105	R/S
106	*LBL
107	D
108	RCL
109	5
110	R/S
111	*LBL
112	E
113	RCL
114	1
115	×
116	RCL
117	2
118	=
119	R/S

Benutzeranleitung:

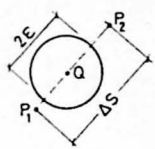
Speicherplan		Eingabe	Taste	Anzeige
T	ϵ	ϵ	x ⇆ t	–
0	Δt	b	R/S	–
1	t	y_0	R/S	–
2	v_P	v_P	R/S	–
3	v_Q, $\Delta\eta$	v_Q	R/S	–
4	b	Δt	R/S	0
5	η	–	A	t
6	x	–	B	x
7	y_0, y	–	C	y
8	$\Delta s/d$	–	D	η
			E	s

Bei Benutzung eines Druckers werden zur Ausgabe von t, x, y d [•Prt] in die PSS 021, 027, 037, 042 und [•Adv] in die PSS 043 gegeben.

Wir führen die Rechnung durch für

$b = 30\,m$, $y_0 = 10\,m$, $v_P = 12\,m/s$, $v_Q = 6\,m/s$

und verschiedene ϵ und Δt. Bei der Festlegung dieser Größen müssen wir $\Delta s = v_P \Delta t \leqq 2\epsilon$ beachten, sonst kommt das Verfahren unter Umständen nicht zum Stillstand. Der Punkt P wird nicht zwangsläufig in die ϵ-Umgebung des Punktes Q gelangen, siehe nebenstehendes Bild mit $\Delta s > 2\epsilon$.



Die Bahnkurve bestimmen wir grob mit $\Delta t = 0{,}25\,s$ und $\epsilon = 1{,}5\,m$ und erhalten vom Drucker die Werte der Tabelle 7.3.2.

Tabelle 7.3.2

0.000	0.750	1.500	2.250
0.000	8.669	17.629	26.092
10.000	7.613	7.088	9.877
31.623	21.557	12.518	5.329
0.250	1.000	1.750	2.500
2.846	11.637	20.594	28.292
9.051	7.180	7.546	11.917
28.184	18.401	9.859	3.525
0.500	1.250	2.000	2.750
5.736	14.631	23.456	29.746
8.248	6.988	8.445	14.541
24.825	15.378	7.447	1.975
			3.000
			30.132
			17.516
			0.502

Hiernach ersetzen wir im Programm die Anweisung [•Prt] und [•Adv] durch [•Nop] und lassen jetzt nur die Endwerte t, x, y, η, s ausrechnen:

Δt/s	ϵ/m	t/s	x/m	y/m	η/m	s/m	Rechenzeit/min
0,1	0,6	3,00	30,004	17,976	18	36	1,7
0,01	0,06	2,96	30,000	17,745	17,76	35,52	17,4
0,001	0,006	2,958	30,000	17,745	17,748	35,496	175

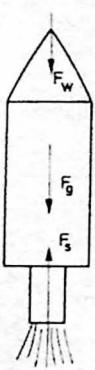
Aus der Angabe der Rechenzeit erkennen Sie, daß wir etwas Geduld beim Warten auf die Ergebnisse haben müssen. Aber wir können diese Zeit anderweitig nutzen, der programmierbare Taschenrechner arbeitet für uns. Beim Wert $\Delta t = 0{,}001$ s durchläuft der Rechner etwa 3000 mal die Schleife im Programm.

Wer übrigens das Problem exakt lösen will, muß die Lösungsfunktion einer Differentialgleichung 2. Ordnung mit Anfangsbedingungen bestimmen und erhält dann

$y = \eta = 17{,}7485$ m; $t = 2{,}9581$ s; $s = 35{,}496$ m.

7.3.3. Bewegung einer Rakete

Eine Rakete wird aus dem Stillstand durch den ausströmenden Treibstoff senkrecht nach oben bewegt. Gegeben:



$m_R = 260$ kg (Masse der Rakete);

$m_T = 240$ kg (Masse des Treibstoffs);

$T = 15$ s (Brenndauer);

$v_r = 2000 \frac{m}{s}$ (relative Ausströmgeschwindigkeit des Treibstoffs);

$c_w = 0{,}18$ (Widerstandsziffer der Rakete);

$\rho_0 = 1{,}293 \frac{kg}{m^3}$ (Dichte der Luft an der Erdoberfläche);

$A = 1{,}62\ m^2$ (Schattenfläche; vom Luftstrom erfaßte Fläche der Rakete);

$g = 9{,}807 \frac{m}{s^2}$ (Erdbeschleunigung; konstant angenommen).

Zu bestimmen sind die Geschwindigkeit $v = v(t)$ und die Höhe $h = h(t)$ für $0 \leqq t \leqq T$, insbesondere $v_e = v(T)$ und $H = h(T)$. Die Bewegungsgleichung der Rakete lautet[1]:

$$m(t) \frac{dv}{dt} = F_S - F_W - F_g$$

mit

Masse $m = m(t) = (m_R + m_T) - \frac{m_T}{T} t = m_0 - \dot{m} t$,

[1] s. Szabó: Einführung in die Technische Mechanik, Springer 1975

Schubkraft $F_S = \dot{m} v_r$;

Gewichtskraft $F_g = m(t)\, g$;

Widerstandskraft[1] $F_W = \frac{c_w}{2} A \rho v^2$ und

Dichte[1] in der Höhe h (in m) $\rho = \rho_0 e^{-0{,}0001\, h}$.

Somit wird

$$\frac{dv}{dt} = \frac{F_S - c\, e^{-0{,}0001\, h} v^2}{m_0 - \dot{m} t} - g$$

mit $m_0 = 500$ kg, $\dot{m} = 16 \frac{kg}{s}$, $F_S = 32\,000$ N, $c = \frac{c_w}{2} A \rho_0 = 0{,}1885 \frac{kg}{m}$.

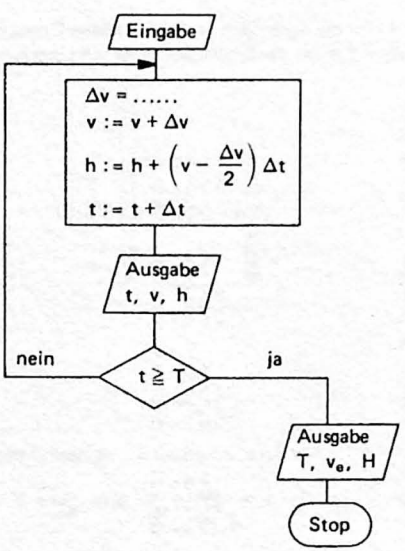
In dieser nicht geschlossen lösbaren Differentialgleichung ersetzen wir $\frac{dv}{dt}$ durch den Differenzenquotienten $\frac{\Delta v}{\Delta t}$ und erhalten:

$$\Delta v = \left(\frac{F_S - c\, e^{-0{,}0001\, h} v^2}{m_0 - \dot{m} t} - g \right) \Delta t ;$$

$$v(t + \Delta t) = v(t) + \Delta v ;$$

$$h(t + \Delta t) = h(t) + v_m \Delta t = h(t) + \left[v(t + \Delta t) - \frac{\Delta v}{2} \right] \Delta t .$$

Wir zeichnen in großen Zügen das *Flußdiagramm:*



Speicherplan	
T	T
0	Δt
1	t
2	F_S
3	c
4	m_0
5	$\dot{m}$
6	g
7	− 0,0001
8	v
9	h

[1] s. Athen: Ballistik, Quelle u. Meyer 1958

Die für die Berechnungen benötigten Werte Δt, F_S, c usw. geben wir manuell nach obigem Speicherplan ein. Als Einheit werden für die Zeit Sekunde, für Längen Meter, für Massen Kilogramm und für Kräfte Newton $(1\,N = 1\,\frac{kg\,m}{s^2})$ benutzt.

Das Programm soll so angelegt werden, daß der Rechner entweder t, v(t), h(t) als Tabelle über einen Drucker ausgibt oder nur die Endwerte T, v_e, H anzeigt.

PSS	Taste
000	RCL
001	2
002	−
003	RCL
004	3
005	x
006	(
007	RCL
008	7
009	x
010	RCL
011	9
012	)
013	INV
014	ln x
015	x
016	RCL
017	8
018	x^2

019	=
020	÷
021	(
022	RCL
023	4
024	−
025	RCL
026	5
027	x
028	RCL
029	1
030	)
031	−
032	RCL
033	6
034	=
035	x
036	RCL
037	0
038	=

039	SUM
040	8
041	+/−
042	÷
043	2
044	+
045	RCL
046	8
047	=
048	x
049	RCL
050	0
051	=
052	SUM
053	9
054	RCL
055	0
056	SUM
057	1
058	RCL

059	1
060	*Prt
061	RCL
062	8
063	*Prt
064	RCL
065	9
066	*Prt
067	*Adv
068	RCL
069	1
070	*x≧t
071	0
072	74
073	RST
074	R/S

In der obigen Form wird das Programm für die Ausgabe t, v, h über einen Drucker benutzt. So wurden die Werte der Tabelle 7.3.3 mit der Schrittweite $\Delta t = 1\,s$ berechnet.

Tabelle 7.3.3

1.0	6.0	11.0
54.2	296.5	411.0
27.1	946.4	2755.6
2.0	7.0	12.0
109.4	328.6	425.3
108.9	1258.9	3173.8
3.0	8.0	13.0
163.2	355.0	438.8
245.1	1600.7	3605.8
4.0	9.0	14.0
213.3	376.8	451.9
433.4	1966.6	4051.2
5.0	10.0	15.0
258.1	395.1	465.0
669.1	2352.5	4509.7

Steht kein Drucker zur Verfügung, so werden die Anweisungen [*Prt] durch [R/S] und [*Adv] durch [*Nop] ersetzt. Die Werte t, v, h müssen dann von der Anzeige abgeschrieben werden.

Interessieren wir uns nur für $v_e = v(T)$ und $H = h(T)$, so werden alle Drucker-Anweisungen durch [*Nop] ersetzt. Wir beachten dann für die

Ausgabe: T [RCL] 08 v_e [RCL] 09 H.

Wir haben die Berechnung von v_e und H für einige Δt durchgeführt:

Δt/s	v_e/(m/s)	H/m
1	465,0	4509,7
0,5	463,9	4478,4
0,1	462,9	4452,1
0,01	462,7	4446,0

7.4. Elektrotechnik

7.4.1. Spannungsteiler

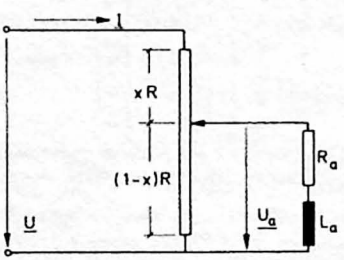


Bild 7.4.1

Für einen Spannungsteiler (Bild 7.4.1) sei der Widerstand R stetig teilbar in x R und (1−x) R mit $0 \leq x \leq 1$. Gegeben sind:

$R = 150\,\Omega$; $R_a = 80\,\Omega$;
$L_a = 0{,}24\,H$; $f = 50\,Hz$.

Das Verhältnis $\frac{\underline{U}_a}{\underline{U}}$ und der Phasenwinkel φ zwischen den Spannungen $\underline{U}$ und $\underline{U}_a$ sollen in Abhängigkeit von x dargestellt und berechnet werden.

Nach den Gesetzen der Elektrotechnik gilt

$$\underline{U} = \underline{I} \cdot R + \underline{U}_a \quad \text{und} \quad \underline{U}_a = \underline{I}\,\underline{Z} \quad \text{mit} \quad \frac{1}{\underline{Z}} = \frac{1}{(1-x)\,R} + \frac{1}{R_a + j\omega L_a} \quad \text{und} \quad \omega = 2\pi f .$$

Hieraus erhalten wir nach etwas längeren algebraischen Umformungen die folgenden Berechnungsformeln:

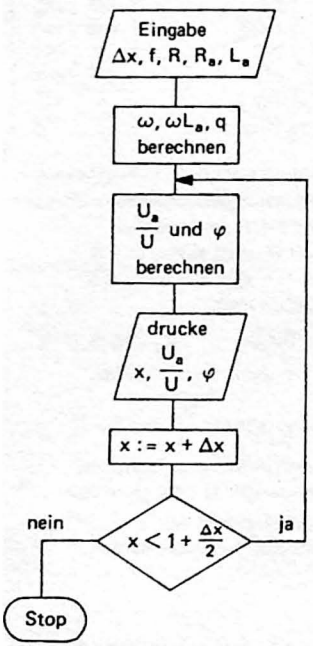
$$q = \frac{R_a^2 + (\omega L_a)^2}{R}; \quad y = x(1-x);$$

$$\frac{U_a}{U} = \frac{1-x}{\sqrt{1 + \frac{y}{q}(2R_a + yR)}};$$

$$\varphi = \arctan \frac{y \omega L_a}{q + y R_a}.$$

Die Funktionswerte $\frac{U_a}{U}$ und φ (im Gradmaß) wollen wir uns tabelliert vom Drucker von $x = 0$ bis $x = 1$ mit der Schrittweite Δx ausgeben lassen.

Flußdiagramm:



Speicherplan	
T	$1 + \frac{\Delta x}{2}$
0	Δx
1	f, ω
2	R
3	R_a
4	$L_a, \omega L_a$
5	q
6	x
7	$1-x, y$
8	Ind

Für die Eingabe der gegebenen Werte wählen wir die indirekte Adressierung mit [A].

PSS	Taste
000	*CM
001	*LBL
002	A
003	STO *Ind
004	08
005	1
006	SUM
007	8
008	R/S
009	*LBL
010	B
011	1
012	+
013	RCL
014	0
015	÷
016	2
017	=
018	x⇄t
019	2
020	×
021	*π
022	=
023	*Prd
024	1
025	RCL
026	1
027	*Prd

028	4
029	RCL
030	3
031	x^2
032	+
033	RCL
034	4
035	x^2
036	=
037	÷
038	RCL
039	2
040	=
041	STO
042	05
043	1
044	−
045	RCL
046	6
047	*Prt
048	=
049	STO
050	7
051	÷
052	(
053	1
054	+
055	RCL
056	6

057	*Prd
058	7
059	RCL
060	7
061	÷
062	RCL
063	5
064	×
065	(
066	2
067	×
068	RCL
069	3
070	+
071	RCL
072	7
073	×
074	RCL
075	2
076	)
077	)
078	$\sqrt{x}$
079	=
080	*Prt
081	RCL
082	7
083	×
084	RCL
085	4

086	=
087	÷
088	(
089	RCL
090	5
091	+
092	RCL
093	7
094	X
095	RCL
096	3
097	)
098	=
099	INV
100	*tan
101	*Prt
102	*Adv
103	RCL
104	0
105	SUM
106	6
107	RCL
108	6
109	INV
110	*x≥t
111	0
112	43
113	R/S
114	RST

Eingabe: Δx [R/S] f [A] R [A] R_a [A] L_a [A];

Ausgabe: [B] für Drucker;

[B] x [R/S] U_a/U [R/S] φ ohne Drucker.

Steht *kein* Drucker zur Verfügung, so sind in diesem Programm die Anweisungen [*Prt] durch [R/S] und [*Adv] durch [*Nop] zu ersetzen.

Die Ergebnisse der Rechnung für die obigen Zahlenwerte des Spannungsteilers sind für $\Delta x = 0{,}1$ in der Tabelle 7.4.1 angegeben. In Bild 7.4.2 sind die Funktionen $\frac{U_a}{U}$ (ausgezogen) und φ (gestrichelt) in Abhängigkeit von x graphisch dargestellt.

Tabelle 7.4.1

0.000	0.300	0.600	0.900
1.000	0.572	0.318	0.092
0.000	9.237	10.281	4.421
0.100	0.400	0.700	1.000
0.824	0.477	0.245	0.000
4.421	10.281	9.237	0.000
0.200	0.500	0.800	
0.685	0.394	0.171	
7.362	10.616	7.362	

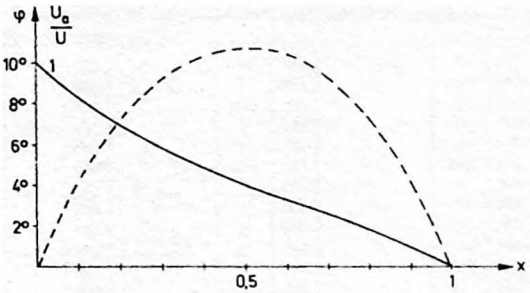


Bild 7.4.2

7.4.2. Berechnung eines Wechselstromkreises

Für den in Bild 7.4.3 dargestellten Wechselstromkreis sind gegeben: Frequenz f, Spannung U, Widerstände bzw. Induktivitäten R_0, L_0, R_1, L_1, R_2. Folgende Größen sollen berechnet werden:

Ersatzwiderstand $\underline{Z} = X + jY$,

Phasenwinkel φ zwischen Spannung $\underline{U}$ und Strom $\underline{I}$,

effektive Stromstärke I.

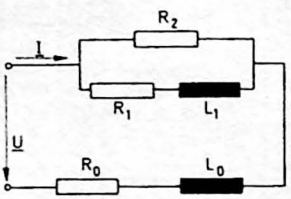


Bild 7.4.3

Es gilt

$$\underline{Z} = R_0 + j\omega L_0 + \underline{Z}_p \quad \text{mit}$$

$$\frac{1}{\underline{Z}_p} = \frac{1}{R_2} + \frac{1}{R_1 + j\omega L_1} \quad \text{und}$$

$$\underline{U} = \underline{I}\,\underline{Z} = \underline{I}\,|\underline{Z}|\,e^{j\varphi}.$$

Hieraus erhalten wir nach einigen Umformungen:

$$X = R_0 + q\,[R_1(R_1 + R_2) + (\omega L_1)^2]$$

$$Y = \omega L_0 + q\,R_2\,\omega L_1$$

$$\varphi = \arctan\frac{Y}{X}$$

$$I = \frac{U}{X}\cos\varphi$$

mit $\omega = 2\pi f$ und $q = \dfrac{R_2}{(R_1 + R_2)^2 + (\omega L_1)^2}$

Wir wollen voraussetzen, daß R_1, L_1, R_2 nicht gleichzeitig Null sind. Sonst wäre $q = \frac{0}{0}$ unbestimmt, und wir müßten Verzweigungen ins Programm hineinnehmen. Auch den Fall $X = 0$ (z.B. $R_0 = R_2 = 0$) wollen wir ausschließen. Diese Fälle werden vom Rechner durch Blinken angezeigt.

Wir geben das Programm zur Berechnung der gesuchten Größen X, Y, φ, I mit ausführlichen Bemerkungen an. Die Datenregister (Speicher) bezeichnen wir hier mit S, um sie nicht mit den Widerständen R zu verwechseln.

Speicherplan	
0	ω
1	U
2	R_0
3	$\omega, \omega L_0$
4	R_1
5	$\omega, \omega L_1$
6	R_2
7	$R_1 + R_2$
8	q
9	X

PSS	Taste	Bemerkungen
000	*CM$_s$	$(S_n) = 0$, Eing. f
001	×	
002	2	
003	×	2 f
004	*π	
005	=	$\omega = 2\pi f$
006	STO	$\omega \rightarrow S_0$
007	0	
008	STO	$\omega \rightarrow S_3$
009	3	
010	STO	$\omega \rightarrow S_5$
011	5	
012	R/S	Eingabe U
013	STO	$U \rightarrow S_1$
014	1	
015	R/S	Eingabe R_0
016	STO	$R_0 \rightarrow S_2$
017	2	
018	R/S	Eingabe L_0
019	*Prd	$\omega L_0 \rightarrow S_3$
020	3	
021	R/S	Eingabe R_1
022	STO	$R_1 \rightarrow S_4$
023	4	
024	STO	$R_1 \rightarrow S_7$
025	7	
026	R/S	Eingabe L_1
027	*Prd	$\omega L_1 \rightarrow S_5$
028	5	
029	R/S	Eingabe R_2
030	STO	$R_2 \rightarrow S_6$
031	6	
032	SUM	$(R_1 + R_2) \rightarrow S_7$
033	7	
034	÷	
035	(	
036	RCL	$(R_1 + R_2) \rightarrow AR$
037	7	
038	x^2	$(R_1 + R_2)^2$
039	+	
040	RCL	$\omega L_1 \rightarrow AR$
041	5	
042	x^2	$(\omega L_1)^2$
043	)	Nenner von q
044	=	q
045	STO	$q \rightarrow S_8$
046	8	
047	×	

048	(	
049	RCL	$R_1 \rightarrow AR$
050	4	
051	×	
052	RCL	$(R_1 + R_2) \rightarrow AR$
053	7	
054	+	$R_1 (R_1 + R_2)$
055	RCL	$\omega L_1 \rightarrow AR$
056	5	
057	x^2	$(\omega L_1)^2$
058	)	[...]
059	+	q [...]
060	RCL	$R_0 \rightarrow AR$
061	2	
062	=	X
063	STO	$X \rightarrow S_9$
064	9	
065	R/S	Anzeige X
066	RCL	$\omega L_0 \rightarrow AR$
067	3	
068	+	
069	RCL	$q \rightarrow AR$
070	8	
071	×	
072	RCL	$R_2 \rightarrow AR$
073	6	
074	×	
075	RCL	$\omega L_1 \rightarrow AR$
076	5	
077	=	Y
078	R/S	Anzeige Y
079	÷	
080	RCL	$X \rightarrow AR$
081	9	
082	=	Y/X
083	INV	
084	*tan	φ
085	R/S	Anzeige φ
086	*cos	$\cos\varphi$
087	×	
088	RCL	$U \rightarrow AR$
089	1	
090	÷	$U \cos\varphi$
091	RCL	$X \rightarrow AR$
092	9	
093	=	*I*
094	R/S	Anzeige *I*
095	RST	Sprung auf 000

Eingabe: f [R/S] U [R/S] R_0 [R/S] L_0 [R/S] R_1 [R/S] L_1 [R/S] R_2 [R/S] ;

Ausgabe: X [R/S] Y [R/S] φ [R/S] I .

In den folgenden Zahlenbeispielen dienen die ersten drei Zeilen zur Kontrolle des Programms. f = 50 Hz; U = 220 V.

R_0/Ω	L_0/H	R_1/Ω	L_1/H	R_2/Ω	X/Ω	Y/Ω	φ	I/A
100	0	1	bel.	0	100	0	0°	2,2
50	0	100	0	100	100	0	0°	2,2
50	$1/\pi$	bel.	bel.	0	50	100	63,4°	1,97
86	0	125	0	150	154,2	0	0°	1,43
110	0,45	65	0,84	120	204,4	178,0	41,1°	0,812
0	0,182	46	0,260	35	27,50	64,74	67,0°	3,13
450	2,5	300	1,8	280	660,7	853,0	52,2°	0,204
320	0	0	1,2	400	508,2	199,6	21,4°	0,403

7.4.3. Harmonische Analyse

In einem Stromkreis sei die Stromstärke i oder die Spannung u eine Funktion der Zeit: y = f(t). Von dieser Funktion wollen wir die folgenden Eigenschaften (Bild 7.4.4) voraussetzen:

1. f(t) sei periodisch mit der Periode T, d.h. f(t + T) = f(t),
2. die Funktionswerte $y_k = f(t_k)$ seien an 12 äquidistanten Stellen $t_k = k\,\Delta t$ $(k \in \mathbb{N}_{12})$ bekannt (z.B. empirisch ermittelt).

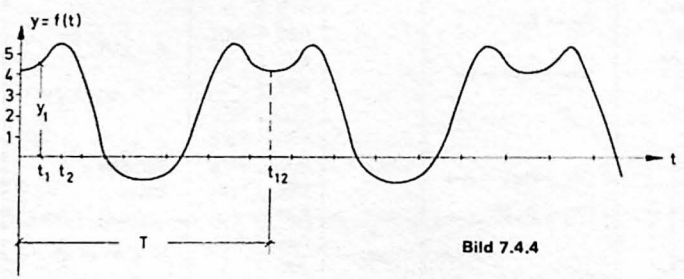


Bild 7.4.4

Die Funktion y = f(t) läßt sich angenähert als Summe trigonometrischer Funktionen darstellen (mit $\omega = \frac{2\pi}{T}$):

$$y = f(t) \approx a_0 + a_1 \cos \omega t + a_2 \cos 2\omega t + \ldots + a_5 \cos 5\omega t + a_6 \cos 6\omega t + b_1 \sin \omega t + b_2 \sin 2\omega t + \ldots + b_5 \sin 5\omega t \,.$$

Zur Berechnung der Fourier-Koeffizienten[1]

$$a_0 = \frac{1}{12}\sum_{k=1}^{12} y_k ; \qquad a_6 = \frac{1}{12}\sum_{k=1}^{12} y_k \cos\left(6\,k\,\frac{\pi}{6}\right) ;$$

$$a_n = \frac{1}{6}\sum_{k=1}^{12} y_k \cos\left(n\,k\,\frac{\pi}{6}\right) \qquad (n \in \mathbb{N}_5)$$

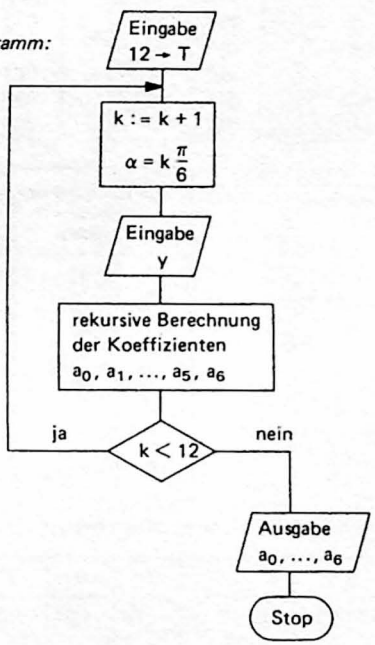
wollen wir ein Programm schreiben. (Das entsprechende Programm für die Koeffizienten $b_n = \frac{1}{6}\sum_{k=1}^{12} y_k \sin\left(n\,k\,\frac{\pi}{6}\right)$ möge der Leser selbst schreiben.)

Mit $\alpha = k\,\frac{\pi}{6}$ berechnen wir die a_k rekursiv wie folgt:

$$a_0 := a_0 + \frac{1}{12}\,y ; \qquad a_6 := a_6 + \frac{1}{12}\,y\cos(6\alpha) ;$$

$$a_n := a_n + \frac{1}{6}\,y\cos(n\alpha) \qquad (n \in \mathbb{N}_5) .$$

Flußdiagramm:



Die häufiger auftretenden Berechnungen von $\frac{1}{6}\,y\cos(n\alpha)$ lassen wir zum Teil in einem Unterprogramm (SBR 074) durchführen. Die Ausgabe der Koeffizienten a_n nehmen wir nicht in das Programm hinein, wir führen sie direkt über RCL on durch.

[1] s. z.B. Zurmühl: Praktische Mathematik f. Ing. u. Phys., Springer 1965

Das Programm lautet:

PSS	Taste
000	*CM$_s$
001	*Rad
002	1
003	2
004	x ≷ t
005	1
006	SUM
007	8
008	RCL
009	8
010	x
011	*π
012	÷
013	6
014	=
015	STO
016	9
017	RCL
018	8
019	R/S
020	STO

021	7
022	÷
023	1
024	2
025	=
026	SUM
027	00
028	1
029	SBR
030	0
031	74
032	SUM
033	01
034	2
035	SBR
036	0
037	74
038	SUM
039	02
040	3
041	SBR
042	0

043	74
044	SUM
045	03
046	4
047	SBR
048	0
049	74
050	SUM
051	04
052	5
053	SBR
054	0
055	74
056	SUM
057	05
058	6
059	SBR
060	0
061	74
062	÷
063	2
064	=

065	SUM
066	6
067	RCL
068	8
069	INV
070	*x ≧ t
071	0
072	05
073	R/S
074	x
075	RCL
076	9
077	=
078	*cos
079	x
080	RCL
081	7
082	÷
083	6
084	=
085	INV SBR

Benutzeranleitung:

Harmonische Analyse: Berechnung der a_n				
Speicherplan		Eingabe	Taste	Anzeige
T	12	–	R/S	1
0	a_0	y_1	R/S	2
1	a_1	y_2	R/S	3
...	...	y_3	...	...
6	a_6	...	...	12
7	y	y_{12}	R/S	13
8	k	–	RCL on	a_n
9	$k\frac{\pi}{6}$			

Für die in Bild 7.4.4 skizzierte Funktion seien die folgenden Funktionswerte gemessen:

k	1	2	3	4	5	6	7	8	9	10	11	12
y_k	4,46	5,21	3,64	0,14	−1,08	−1,24	−1,08	0,14	3,64	5,21	4,46	4,10

Wir haben es hier mit einer geraden Funktion zu tun, d.h. $f(-t) = f(t) = f(T-t)$, daher sind die Koeffizienten $b_n = 0$. Für die a_n erhalten wir mit dem obigen Programm:

n	0	1	2	3	4	5	6
a_n	2,300	3,334	−1,065	−0,800	0,235	0,136	−0,040

8. Anhang

8.1. Tastenkode-Tabelle

In der folgenden Tabelle sind nur die Tasten aufgeführt, die in diesem Buch erklärt oder benutzt wurden.

Kode	Taste	Kode	Taste	Kode	Taste	Kode	Taste
00	0	22	INV	49	*Prd	73	RCL *Ind
01	1	23	ln x	50	*|x|	74	SUM *Ind
02	2	24	CE	52	EE	75	−
03	3	25	CLR	53	(	76	*LBL
04	4	28	*log	54	)	77	*x ≷ t
05	5	29	*CP	55	÷	80	*Grad
06	6	30	*tan	58	*Fix	81	RST
07	7	32	x ▶ t	59	*Int	85	+
08	8	33	x^2	60	*Deg	86	*St flg
09	9	34	$\sqrt{x}$	61	GTO	87	*If flg
10	*E'	35	1/x	63	*Exc *Ind	89	*π
11	A	38	*sin	64	*Prd *Ind	90	*List
12	B	39	*cos	65	×	91	R/S
13	C	40	*Ind	66	*Pause	92	INV SBR
14	D	42	STO	67	*x = t	93	.
15	E	43	RCL	68	*Nop	94	+/−
16	*A'	44	SUM	69	*Op	95	=
17	*B'	45	y^x	70	*Rad	97	*Dsz
18	*C'	47	*CM_s	71	SBR	98	*Adv
19	*D'	48	*Exc	72	STO *Ind	99	*Prt

8.2. Lösungen der Übungsaufgaben

Abschnitt 1.

1.1. a) 0,7957844716 b) −0,2038504417
c) 4767,18188 d) 0,0211319338
e) $-7{,}8492862 \cdot 10^{15}$ f) 0,0814529369
g) 2,994679658 h) 1,975195403

1.2. a) 2 b) 9 c) 1

1.3. $s_1 = 6$; $s_2 = 84$; $p = 4096$; $q = 1$; $r = 2016$.

1.4. $A_M = 443{,}9\ dm^2$; $A = 709{,}8\ dm^2$; $V = 1090{,}2\ dm^3$.

2.1.

PSS	Taste
000	STO
001	1
002	x^2
003	×
004	•π
005	÷
006	4
007	=

008	R/S
009	×
010	RCL
011	1
012	÷
013	8
014	=
015	R/S
016	×

017	RCL
018	1
019	÷
020	2
021	=
022	R/S
023	·
024	5
025	SUM

026	1
027	RCL
028	1
029	GTO
030	0
031	02

d/mm	20	25	30	35	40	45	50
A/cm^2	3,14	4,91	7,07	9,62	12,57	15,90	19,63
W/cm^3	0,785	1,53	2,65	4,21	6,28	8,95	12,27
I/cm^4	0,785	1,92	3,98	7,37	12,57	20,13	30,68

2.2.

PSS	Taste
000	STO
001	1
002	R/S
003	STO
004	2
005	R/S
006	STO
007	3

008	•cos
009	×
010	RCL
011	1
012	×
013	RCL
014	2
015	×
016	2

017	=
018	+/−
019	+
020	RCL
021	1
022	x^2
023	+
024	RCL
025	2

026	x^2
027	=
028	$\sqrt{x}$
029	R/S
030	GTO
031	0
032	06

γ	17,8°	48,9°	81,0°	104,7°	126,5°	146,1°
c/cm	23,2	42,5	63,0	75,7	84,9	90,6

2.3.

PSS	Taste
000	•LBL
001	A
002	STO
003	1
004	R/S

005	•LBL
006	B
007	STO
008	2
009	×
010	2

011	+
012	RCL
013	1
014	÷
015	RCL
016	2

017	x^2
018	=
019	÷
020	3
021	=
022	R/S

2.4.

PSS	Taste
000	STO
001	0
002	R/S
003	STO
004	1
005	R/S
006	÷
007	1
008	0
009	0
010	+
011	1
012	=
013	STO
014	2
015	y^x
016	RCL
017	1
018	=
019	−
020	1
021	=
022	×
023	RCL
024	2
025	÷
026	(
027	RCL
028	2
029	−
030	1
031	)
032	=
033	×
034	RCL
035	0
036	=
037	GTO
038	0
039	05

Ergebnisse für n = 6:

p	5,0	5,25	5,5	5,75	6,0	6,25	6,5
K_6	7142,01	7204,20	7266,89	7330,11	7393,84	7458,09	7522,87

2.5.

PSS	Taste
000	*LBL
001	A
002	STO
003	1
004	R/S
005	*LBL
006	B
007	STO
008	2
009	R/S
010	*LBL
011	C
012	STO
013	3
014	R/S
015	*LBL
016	D
017	RCL
018	2
019	x^2
020	−
021	4
022	×
023	RCL
024	1
025	×
026	RCL
027	3
028	=
029	$\sqrt{x}$
030	STO
031	4
032	−
033	RCL
034	2
035	=
036	÷
037	2
038	÷
039	RCL
040	1
041	=
042	R/S
043	*LBL
044	E
045	RCL
046	4
047	+/−
048	GTO
049	0
050	32

Ergebnisse (auf 4 Nachkommastellen):

x_1	3	2	− 0,3395	27,1146	0,0394
x_2	− 2	0,3333	− 4,8721	4,1454	− 0,0853

2.6.

PSS	Taste
000	STO
001	1
002	R/S
003	STO
004	2
005	R/S
006	STO
007	3
008	.
009	5
010	SUM
011	4
012	RCL
013	4
014	R/S
015	×
016	RCL
017	2
018	=
019	+/−
020	INV
021	ln x
022	×
023	RCL
024	1
025	×
026	(
027	RCL
028	3
029	×
030	RCL
031	4
032	)
033	*sin
034	=
035	R/S
036	GTO
037	0
038	08

t/s	0	0,5	1,0	1,5	2,0	2,5	3,0	3,5
y/mm	0	15,995	20,468	13,096	0,000	−10,722	−13,720	−8,778

4,0	4,5	5,0	5,5	6,0	6,5	7,0	7,5	8,0
0,000	7,187	9,197	5,884	0,000	−4,818	−6,165	−3,944	0,000

2.7. Berechnet wird $a + \sqrt{b^2 + 1} - \frac{2ab}{a+b}$.

Abschnitt 3.

3.1.

PSS	Taste
000	2
001	x ⇆ t
002	R/S
003	STO
004	1
005	*x ≧ t
006	0

007	15
008	x^2
009	÷
010	4
011	=
012	GTO
013	0
014	02

015	×
016	2
017	−
018	1
019	=
020	÷
021	(
022	RCL

023	1
024	+
025	1
026	)
027	=
028	GTO
029	0
030	02

x	0,5	4	1	2	0	3
y	0,0625	1,4	0,25	1	0	1,25

3.2.

PSS	Taste
000	*CM_s
001	STO
002	0
003	STO
004	1
005	R/S
006	STO
007	2
008	R/S
009	+
010	RCL
011	2

012	=
013	×
014	R/S
015	x^2
016	=
017	SUM
018	3
019	*Dsz
020	0
021	0
022	08
023	RCL
024	3

025	R/S
026	$\sqrt{x}$
027	+
028	RCL
029	2
030	x^2
031	=
032	$\sqrt{x}$
033	R/S
034	RCL
035	2
036	×
037	RCL

038	3
039	÷
040	(
041	1
042	+
043	RCL
044	1
045	)
046	=
047	R/S

Ergebnisse:

s = 335,07; r = 4,6813; p = 90,947.

3.3.

PSS	Taste
000	*LBL
001	B
002	STO
003	00
004	1
005	STO
006	2
007	R/S
008	*LBL
009	A
010	STO
011	01
012	1
013	SUM
014	02
015	3
016	÷
017	RCL
018	1
019	+
020	RCL
021	1
022	$\sqrt{x}$
023	÷
024	RCL
025	2
026	=
027	*Dsz
028	0
029	A
030	R/S

	$a_0 = 1$	$a_0 = 2$	$a_0 = 10$
n = 5	2,093889697	1,880055440	1,794906931
n = 20	1,709365334	1,753945899	1,773801058
n = 100	1,729532890	1,737022586	1,74030571

3.4 a

PSS	Taste
000	*CM_s
001	x ▶ t
002	1
003	+/−
004	STO
005	01
006	1
007	+/−
008	*Prd
009	01
010	1
011	SUM
012	00
013	4
014	÷
015	(
016	2
017	x
018	RCL
019	0
020	−
021	1
022	)
023	=
024	x
025	RCL
026	1
027	=
028	SUM
029	2
030	*|x|
031	*x ≧ t
032	0
033	06
034	RCL
035	2
036	R/S
037	RCL
038	0
039	R/S
040	RST

3.4 b

PSS	Taste
000	*CM_s
001	x ▶ t
002	3
003	STO
004	2
005	STO
006	03
007	1
008	SUM
009	1
010	RCL
011	1
012	x
013	2
014	−
015	1
016	=
017	x^2
018	÷
019	(
020	2
021	x
022	RCL
023	1
024	+
025	1
026	)
027	÷
028	RCL
029	1
030	÷
031	8
032	=
033	*Prd
034	2
035	RCL
036	2
037	SUM
038	3
039	*x ≧ t
040	0
041	07
042	RCL
043	3
044	R/S
045	RCL
046	1
047	R/S
048	RST

	ϵ	s	n
3.4 a	0,01	3,146567747	201
3.4 b	0,000001	3,141592511	8

3.5.

PSS	Taste
000	*CM$_s$
001	x ⇄ t
002	R/S
003	*LBL
004	A
005	STO
006	0
007	STO
008	1
009	R/S
010	*LBL
011	B
012	STO
013	3
014	x ⇄ t

015	*x ≧ t
016	0
017	20
018	x ⇄ t
019	*A′
020	x ⇄ t
021	x^2
022	+
023	2
024	·
025	1
026	=
027	÷
028	(
029	·
030	8

031	+
032	RCL
033	3
034	)
035	=
036	SUM
037	4
038	*LBL
039	*A′
040	RCL
041	4
042	*Dsz
043	0
044	0
045	09
046	R/S

047	*LBL
048	C
049	x ⇄ t
050	+
051	RCL
052	4
053	$\sqrt{x}$
054	=
055	÷
056	RCL
057	1
058	=
059	R/S
060	RST
061	
062	

m	1	2	3	4	5	6
s_m	0	2,785	4,657	4,657	7,994	11,030

c = 1,137.

3.6.

PSS	Taste
000	STO
001	0
002	STO
003	01
004	1
005	STO
006	2
007	R/S
008	*LBL
009	A
010	*If flg
011	1
012	*Prd

013	*Prd
014	2
015	*St flg
016	1
017	D
018	*LBL
019	*Prd
020	x^2
021	*Prd
022	2
023	INV
024	*St flg
025	1
026	*LBL

027	D
028	*Dsz
029	0
030	0
031	07
032	R/S
033	*LBL
034	B
035	RCL
036	2
037	R/S
038	*LBL
039	C
040	1

041	+
042	RCL
043	2
044	*|x|
045	=
046	INV
047	y^x
048	RCL
049	1
050	=
051	R/S
052	
053	
054	

p = 347,573; q = 2,653.

3.7. Berechnet wird $s = \sum_{k=1}^{n} (2k-1)^2$.

Abschnitt 4.

4.1.

PSS	Taste
000	*CM$_s$
001	STO
002	00
003	1
004	SUM
005	1

006	SBR
007	0
008	25
009	x^2
010	÷
011	2
012	+

013	1
014	=
015	SUM
016	2
017	*Dsz
018	0
019	0

020	03
021	RCL
022	2
023	R/S
024	RST
025	

s_6 = 6,2039.

4.2.

PSS	Taste
000	*LBL
001	A
002	STO
003	1
004	R/S
005	*LBL
006	B
007	STO
008	2
009	R/S
010	*LBL
011	C
012	SBR
013	$\sqrt{x}$
014	×
015	2
016	=
017	STO

018	03
019	1
020	SUM
021	1
022	SBR
023	$\sqrt{x}$
024	STO
025	5
026	÷
027	3
028	=
029	SUM
030	03
031	1
032	SUM
033	2
034	SBR
035	$\sqrt{x}$
036	STO

037	04
038	1
039	+
040	RCL
041	5
042	=
043	ln x
044	SUM
045	4
046	RCL
047	3
048	STO
049	1
050	R/S
051	*LBL
052	D
053	RCL
054	4
055	STO

056	2
057	R/S
058	*LBL
059	E
060	SBR
061	$\sqrt{x}$
062	R/S
063	*LBL
064	$\sqrt{x}$
065	RCL
066	1
067	x^2
068	+
069	RCL
070	2
071	x^2
072	=
073	$\sqrt{x}$
074	INV SBR

z_1	0,333	2,471	3,162	7,366	10,530	2,230
z_2	2,107	3,117	2,693	5,980	6,242	2,464
z_3	2,134	3,978	4,153	9,488	12,241	3,323

4.3.

PSS	Taste
000	*CM_s
001	STO
002	0
003	STO
004	01
005	1
006	SUM
007	2
008	SBR
009	0
010	45

011	×
012	(
013	1
014	+
015	SBR
016	1
017	50
018	÷
019	RCL
020	1
021	)
022	=

023	SUM
024	3
025	*Dsz
026	0
027	0
028	05
029	RCL
030	3
031	R/S
032	÷
033	RCL
034	1

035	÷
036	(
037	RCL
038	1
039	+
040	1
041	)
042	=
043	R/S
044	RST
045	

n	5	10	20
s	2,050775789	2,727162490	3,185142171
c	0,068359193	0,024792386	0,007583672

4.4.

PSS	Taste
000	*LBL
001	A
002	*CM$_s$
003	STO
004	1
005	R/S
006	*LBL
007	B
008	÷
009	2
010	=
011	STO
012	2
013	×
014	3

015	=
016	INV
017	SUM
018	01
019	0
020	SBR
021	C
022	SBR
023	C
024	×
025	2
026	=
027	SBR
028	C
029	×
030	3

031	=
032	SBR
033	C
034	×
035	2
036	=
037	SBR
038	C
039	SUM
040	3
041	RCL
042	3
043	÷
044	9
045	=
046	R/S

047	*LBL
048	C
049	SUM
050	3
051	RCL
052	2
053	SUM
054	1
055	SBR
056	E
057	INV SBR
058	*LBL
059	E
060	
061	
062	

y_m	2,5230	1,7618	1,6490

4.5. Berechnet wird

$$y = e^{x^2+1} + \sqrt{(x-1)^2+1} + \ln[(x+2)^2+1] + \frac{1}{(x+1)^2+1} .$$

Sachwortverzeichnis

Advance 77
Algebraisches Operations System 3
Anfangswertaufgabe 107 ff.
Anzeigeregister 1
Arcusfunktion 12
Auflisten eines Programms 76
Auflisten von Daten 77

Back Step 19
Bahnkurve 127
Befehlsadreßregister 15
Benutzeranleitung 25
Besselsche Funktionen 95, 107
Betriebsart RECHNEN, LEARN
oder RUN 15
Biegemoment 120
Binomialkoeffizient 84
Bogenmaß 12

Clear 2
– entry 2
– all Memories 5
– Program 17
Cosinussatz 85 ff.

Datenspeicher, -register 5, 16
Decrement and Skip on Zero 49 ff.
Degree 12
Delete 20
Differentialgleichungen 107 ff.
Differenzenschemaverfahren bei
Differentialgleichungen 109
Dreiecksberechnungen 31, 85
Drucker 75 ff.

Eingabe eines Programms 16
Eingabefehler 20 ff.
Enter Exponent 3
Ergibtanweisung, -zeichen 18
Euklidischer Algorithmus 82
Exchange 5
Exponentialform 3
Exponentialfunktion 11
Fakultät 84
Festpunktdarstellung 2
Flächenmomente 117
Flags, Flaggen 53 ff.
Fließpunktdarstellung 2
Flußdiagramm 22
Fourier-Koeffizienten 139
Funktionswerte, Berechnung 90 ff.
Funktionstasten 10 ff.

Gleichungen 97 ff.
Gleitpunktdarstellung 3
Go To 31

Größter gemeinsamer Teiler 82
Gradmaß 12
Grundrechenarten 1

Harmonische Analyse 138
Hierarchie 3, 10
Horner Schema 90

Indirekte Speicheradressierung 28 ff.
Insert 21
Integer 82
Integration, numerische 102 ff.
Iterationsverfahren 7, 98

Korrigieren eines Programms 19 ff.
Kräfte beim Balken 114, 120
Kräfte in der Ebene am Punkt 112
Kurbeltrieb 123
Kurzformadressierung 6

Label 22, 32
Logarithmusfunktion 11
Löschen, s. Clear

Marken 22, 32
Maximum einer Funktion 92
Memories 5

Nachkommastellen 2
Newtonsches Verfahren 101
No Operation 21
Nullstellen von Funktionen 97 ff.

Polygonzugverfahren, verbessertes 107
Polynom 90
Potenzen 10
Präfixtaste 2
Print 77
Programmablaufplan 22
Programmadreßtasten 22 ff.
Programmspeicherstelle 15
Pythagoräische Zahlen 81

Quadratwurzelberechnung 7, 25, 39
Querkraft 120

Radiant 12
Rakete, Bewegung einer 130
Rechenoperationen pro Sekunde 50
Rechenregister 1 f.
Reset 15
Run, Stop 16

scientific notation 3
Signale 53 ff.
Simpsonregel 105

Single Step 19
Sinussatz 31, 85, 88
Spannungsteiler 133
Speicheradressierung, indirekte 28 ff.
Speichertasten 5
Sprunganweisung, bedingte 37 ff.
–, unbedingte 31 ff.
Subroutine 59 ff.

Tastenkode 15
Tastenkodetabelle 141
Termberechnungen 3
Trigonometrische Funktionen 11
Trace 77
Trapezregel 103
T-Speicher, T-Register 40

Überprüfen eines Programms 19
Unterprogramm 59 ff.

Verarbeitungsregister 1 f.
Vergleiche 37
Verminderung und Überspringen bei Null 47 ff.
Verzweigungen 37

Wechselstromkreis 136
Winkelmaße 12

Zahlendarstellungen 2
Zahlenfolge, rekursive 47

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Gloistehn, Hans Heinrich
Programmieren von Taschenrechnern. – Braunschweig: Vieweg.
3. Lehr- und Übungsbuch für den TI-58 und TI-59.
– 1. Aufl. – 1978.
ISBN 3-528-04095-5

1978

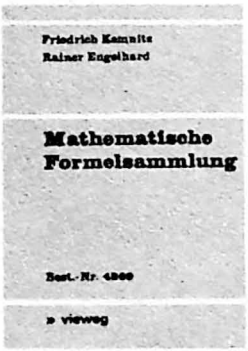
Alle Rechte vorbehalten
© Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig, 1978

Die Vervielfältigung und Übertragung einzelner Textabschnitte, Zeichnungen oder Bilder, auch für Zwecke der Unterrichtsgestaltung, gestattet das Urheberrecht nur, wenn sie mit dem Verlag vorher vereinbart wurden. Im Einzelfall muß über die Zahlung einer Gebühr für die Nutzung fremden geistigen Eigentums entschieden werden. Das gilt für die Vervielfältigung durch alle Verfahren, einschließlich Speicherung und jede Übertragung auf Papier, Transparente, Filme, Bänder, Platten und andere Medien.

Satz: Vieweg, Wiesbaden
Druck: E. Hunold, Braunschweig
Buchbinderische Verarbeitung: W. Langelüddecke, Braunschweig
Printed in Germany West

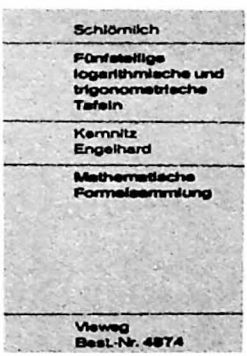
ISBN 3 528 04095 5

Vieweg-Tafeln
und
Mathematische Formelsammlungen



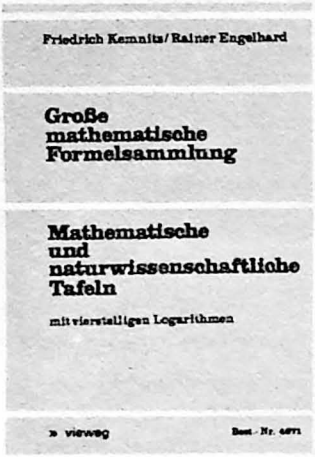
Friedrich Kemnitz und Rainer Engelhard
Mathematische Formelsammlung
1977. 64 Seiten. 12 × 16,5 cm. Kartoniert
ISBN 3 528 04869 7

Oskar Schlömilch
Fünfstellige logarithmische und trigonometrische Tafeln
62., durchgesehene und ergänzte Auflage
1976. II, 176 Seiten. DIN A 6. Gebunden
ISBN 3 528 24873 4



Oskar Schlömilch, Friedrich Kemnitz und Rainer Engelhard
Fünfstellige logarithmische und trigonometrische Tafeln
— Mathematische Formelsammlung
Die Schlömilch'schen Logarithmen (62., durchgesehene Auflage) mit der Mathematischen Formelsammlung (3., durchgesehene Auflage) in einem Band. 1976. II, 176 Seiten und 64 Seiten. DIN A 6. Gebunden
ISBN 3 528 24874 2

Vieweg-Tafeln
und
Mathematische Formelsammlungen



Friedrich Kemnitz und
Rainer Engelhard

Große mathematische Formelsammlung – Mathematisch und naturwissenschaftliche Tafeln

mit vierstelligen Logarithmen. 1975.
64 Seiten und 49 Seiten, DIN C 5.
Kartoniert
ISBN 3 528 14871 3

Friedrich Kemnitz und
Rainer Engelhard

Mathematische und naturwissenschaftliche Tafeln mit vierstelligen Logarithmen -- Kleine mathematische Formelsammlung

1977. 48 Seiten und 24 Seiten. DIN C 5.
Kartoniert
ISBN 3 528 04868 9

In der Reihe

„Programmieren von Taschenrechnern"

von Hans Heinrich Gloistehn
sind erschienen:

Band 1 Lehr- und Übungsbuch für den SR 56

Band 2 Lehr- und Übungsbuch für den TI-57

Band 3 Lehr- und Übungsbuch für den TI-58 und TI-59

Weitere Literatur für Benutzer Elektronischer Taschenrechner:

Taschenrechner Handbuch – Naturwissenschaften / Technik, von H. Schumny

Allgemeine Anleitung zur Auswahl und Gebrauch elektronischer Taschenrechner.

Elektronische Taschenrechner in der Schule, von A. Wynands und U. Wynands

Ein Arbeits- und Aufgabenbuch für Lehrer und Schüler.

Differentialgleichungen der Elektrotechnik, von K. Hoyer und G. Schnell

An Beispielen aus der Elektrotechnik wird neben den klassischen Lösungsverfahren besonders die numerische Berechnung mit programmierbaren Taschenrechnern dargestellt.

Vieweg

Hans Heinrich Gloistehn

Programmieren von Taschenrechnern 3

Lehr- und Übungsbuch für den TI-58 und TI-59

Dieses Lehr- und Übungsbuch führt den Leser in das Programmieren von Taschenrechnern ein, ohne daß Kenntnisse auf diesem Gebiet vorausgesetzt werden. Der Leser lernt die Programmiertechnik und die Fähigkeit, mathematische und technische Probleme zu formulieren und in die Sprache des Rechners umzusetzen.

Im I. Teil des Buches werden die folgenden Gebiete behandelt:

- Manuelles Rechnen
- Programmaufbau und Programmherstellung
- Verzweigungen
- Unterprogramme
- Der Drucker PC-100 A

Im II. Teil wird anhand vieler Beispiele aus der Mathematik und Technik gezeigt, wie die Programmiertechnik angewendet wird.

Zahlreiche Übungsaufgaben geben dem Leser die Gelegenheit, sein gelerntes Wissen zu überprüfen oder zu festigen.

Das Buch wendet sich vorwiegend an Studenten an Fachhochschulen und Universitäten und an Lehrer und Schüler der Sekundarstufe II.

Dr. rer. nat. Hans Heinrich Gloistehn ist Dozent für die Lehrgebiete Mathematik und Technische Mechanik an der Fachhochschule Hamburg.

Best.-Nr. 295/73806